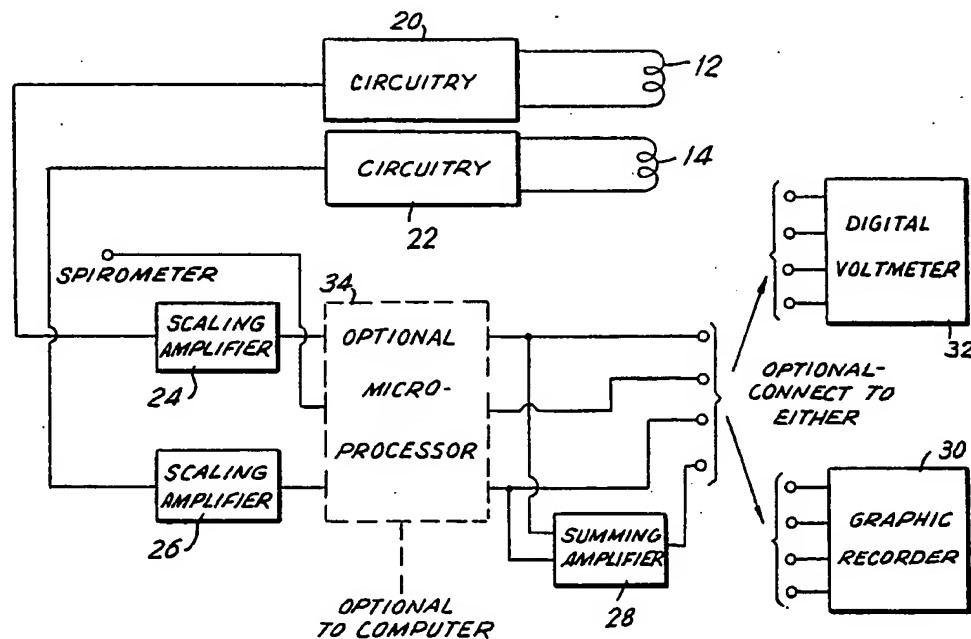




INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification ⁴ : A61B 5/08	A1	(11) International Publication Number: WO 87/ 04332 (43) International Publication Date: 30 July 1987 (30.07.87)
(21) International Application Number: PCT/US87/00217 (22) International Filing Date: 21 January 1987 (21.01.87) (31) Priority Application Number: 820,604 (32) Priority Date: 21 January 1986 (21.01.86) (33) Priority Country: US (71) Applicant: RESPITRACE CORPORATION [US/US]; 1842 West Avenue, Miami, FL 33139-1432 (US). (72) Inventor: WATSON, Herman ; 1842 West Avenue, Miami, FL 33139-1432 (US). (74) Agents: PAVANE, Martin et al.; Stiefel, Gross, Kur- land & Pavane, 551 Fifth Avenue, New York, NY 10176 (US).		(81) Designated States: AT (European patent), AU, BE (Eu- ropean patent), BR, CH (European patent), DE (Eu- ropean patent), DK, FI, FR (European patent), GB (European patent), IT (European patent), JP, KR, LU (European patent), NL (European patent), NO, SE (European patent), SU. Published <i>With international search report.</i>

(54) Title: SINGLE POSITION NON-INVASIVE CALIBRATION TECHNIQUE FOR RESPIRATION MONITORING APPARATUS



(57) Abstract

Method for non-invasively measuring respiration volume whereby weighting factors are non-invasively determined by (a) totaling delta values over a baseline period of substantially steady state breathing for obtaining a rib cage signal, (b) totaling delta values over a baseline period of substantially steady state breathing for obtaining an abdominal signal, (c) dividing the average variability of the mean of the total of the delta values for one of either the rib cage or abdomen signals by the average variability of the mean of the total of the delta values for the other of either the rib cage or abdomen signals; and (d) multiplying the other signal by the quotient derived from step (c).

FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AT	Austria	FR	France	ML	Mali
AU	Australia	GA	Gabon	MR	Mauritania
BB	Barbados	GB	United Kingdom	MW	Malawi
BE	Belgium	HU	Hungary	NL	Netherlands
BG	Bulgaria	IT	Italy	NO	Norway
BJ	Benin	JP	Japan	RO	Romania
BR	Brazil	KP	Democratic People's Republic of Korea	SD	Sudan
CF	Central African Republic	KR	Republic of Korea	SE	Sweden
CG	Congo	LI	Liechtenstein	SN	Senegal
CH	Switzerland	LK	Sri Lanka	SU	Soviet Union
CM	Cameroon	LU	Luxembourg	TD	Chad
DE	Germany, Federal Republic of	MC	Monaco	TG	Togo
DK	Denmark	MG	Madagascar	US	United States of America
FI	Finland				

SINGLE POSITION NON-INVASIVE CALIBRATION TECHNIQUE FOR
RESPIRATION MONITORING APPARATUS

Technical Field

This invention pertains to methods and apparatus for measuring respiration volume and, more particularly, to such methods and apparatus which measure respiration volume by separately measuring and then summing the contributions from a plurality of torso portions, such as the rib cage and abdomen. Most particularly, this invention pertains to a calibration technique for weighting signals indicative of the contributions from the torso portions whereby the sum of the signals is proportional to respiration volume.

Background Art

United States Patent No. 4,308,872 of January 5, 1982, entitled Method And Apparatus For Monitoring Respiration, the contents of which are incorporated herein by reference in their entirety, discloses a method and apparatus for quantitatively measuring respiration volume. The method disclosed in that patent comprises looping first and second extensible conductors about the rib cage and abdomen, separately and simultaneously measuring the inductances of the conductors during respiration, weighting the measured inductances to reflect the different contributions of the rib cage and abdomen to respiration volume, and summing the weighted measured inductances to obtain actual respiration volume.

As noted, practice of the technique disclosed in the patent requires weighting or calibrating the inductances measured by the abdomen and rib cage conductors. To effect calibration it is necessary to determine the weighting factors K and L to satisfy the following equation:

$$V = K \cdot RC + L \cdot AB \quad \text{[EQUATION A]}$$

where V is total respiration volume, RC is the rib cage

BAD ORIGINAL

-2-

1 contribution to respiration volume as measured at the rib
cage conductor and AB is the abdominal contribution as
measured at the abdominal conductor. Patent No. 4,308,872
5 discloses a specific technique for determining the values
for the weighting factors K and L.

In accordance with that disclosure, a spirometer is
employed during the calibration procedure. With the
10 patient in a first position, such as standing, a simulta-
neous set of readings are recorded from the outputs of the
spirometer, the rib cage conductor, and the abdominal con-
15 ductor. This is repeated with the patient in a second
position such as supine. At this point, there are two
sets of values for V, RC, and AB which satisfy Equation A.
20 Thus, two equations having two unknowns, the constants K
and L, may be written. From these, the weighting factors
K and L may be determined by employing well known
techniques for solving simultaneous equations. Thus:

$$25 \quad K = \frac{AB_1 \cdot V_2 - AB_2 \cdot V_1}{RC_2 \cdot AB_1 - RC_1 \cdot AB_2} \quad \text{(EQUATION B)}$$

$$30 \quad L = \frac{RC_1 \cdot V_2 - V_1 \cdot RC_2}{AB_2 \cdot RC_1 - AB_1 \cdot RC_2} \quad \text{(EQUATION C)}$$

35 The denominators of Equations B and C may, depending
upon the recorded values, approach or equal zero.
Clearly, when this happens, the values obtained for K and
L will be inaccurate, thereby skewing any measurement
40 based on such weighting factors. Thus, each time the
denominators of Equations B and C approach or equal zero,
a new set of readings must be taken, thereby increasing
45 the time required for calibration.

In U.S. Patent No. 4,373,534 of February 15, 1983,
entitled Method And Apparatus For Calibrating Respiration
50 Monitoring System, the contents of which are also incor-
porated herein by reference in their entirety, an alter-
nate method and apparatus for a graphing-based technique
for determining the weighting factors K and L is
55 disclosed. As in the simultaneous equation technique of

BAD ORIGINAL

-3-

1 U.S. Patent No. 4,308,872, a spirometer or other device
for independently measuring respiration volume is employed
during the calibration procedure. With the subject in a
5 first position, readings from the spirometer, the rib cage
conductor and the abdominal conductor are simultaneously
recorded for a plurality of breaths, preferably at least
three in number. This is repeated with the subject in a
10 second position. For each breath, the rib cage and abdom-
inal readings are divided by the spirometer reading. That
is, the values RC/V and AB/V are obtained for each breath,
15 where V is the respiration volume as measured by the
spirometer, RC is the rib cage reading from the uncali-
brated rib cage conductor, and AB is the abdominal reading
from the uncalibrated abdominal conductor. The points
20 (RC/D , AB/V) for each breath are next plotted on a graph
whose axes are RC/V and AB/V , and a line approximation is
drawn through these points. The line may be drawn by vis-
25 ual approximation, although preferably it is determined by
the least squares technique. The line is then extended
through the x and y axes. The reciprocals of the x and y
30 intercepts define the weighting factors K and L ; i.e. the
reciprocal of the intercept of the RC/V axis defines the
weighting factor K for the rib cage and the reciprocal of
the intercept of the AB/V axis defines the weighting fac-
35 tor L for the abdomen. Preferably, all of the foregoing
calculations are carried out by a microprocessor or other
data processor which performs the calculations and yields
40 values for the weighting factors K and L .

A drawback of the methods and apparatus disclosed in
45 U.S. Patents Nos. 4,308,872 and 4,373,534 is the require-
ment that sets of data points or values -- from the abdom-
inal and rib cage conductors and from the spirometer or
other respiration volume measurement device -- be obtained
50 with the subject for two different distributions of venti-
lation -- i.e. in two separate positions. Where a sub-
ject's physiological condition prevents or dictates
55 against movement from one position to another, however,
calibration of the rib cage and abdominal conductor

BAD ORIGINAL

1 contributions cannot be readily carried out in accordance
with the known methods.

Another drawback of these prior art methods and appara-
tus is that in certain applications, such as neo-natal
5 monitoring, it is not practical to calibrate the apparatus
using an independent respiration volume measuring device
such as a spirometer. For example, since the above
10 calibration techniques require airway connection to a
spirometer or other similar device, significant time is
required to carry out the procedure. This is often
15 unacceptable to new-born nursery staff where time is at a
premium.

Yet another problem with these calibration
20 techniques and apparatus is that they rely on the
assumption that all air movement in the respiratory system
is between the rib cage and spirometer or the abdomen and
the spirometer. In fact, there also exists movement of
25 air between the rib cage and the abdomen during normal
respiration. This RC-AB exchange of air is pendelluft
that occurs continuously and with varying degree in respi-
30 ration. The known methods do not incorporate an allowance
for pendelluft.

35

40

45

50

55

60

1

-5-

5 Another known quantitative calibration technique is described in
Single Position Tidal Breathing Calibration of the Respiratory
Inductive Plethysmograph, Watson et al, Amer. Rev of Respiratory
10 Diseases, Vol. 129, p. A256 (1984). According to that
technique, with the subject in a single position or posture,
15 readings from a spirometer (SP), the rib cage conductor (RC),
and the abdominal conductor (AB) are simultaneously recorded for
a predetermined period of preferably at least one full breath.
20 The curves (SP, RC) and (SP, AB) are then plotted from the
recorded data and each resulting curve is closed by a straight
25 line connecting its beginning and end point. The resulting loop
areas are then calculated as by integration. Then, using any
selected data points simultaneously recorded from the spirometer
30 (SP) and from the rib cage (RC) and abdominal (AB) conductors,
weighting factors (for the rib cage scaling amplifier and/or for
35 the abdomen scaling amplifier) may be determined. This
calibration technique also suffers from the drawback that it is
invasive in the sense that it requires a spirometer or other
40 device measuring respiration volume at the mouth.

45

50

55

60

1

-6-

5 Brief Description Of The Drawings

10 In the drawings, wherein like reference numerals denote similar elements throughout the several views:

FIG. 1 is a diagrammatic representation of a portion of a system for non-invasively monitoring respiration volume;

15 FIG. 2 is a block diagram of a complete system for non-invasively monitoring respiration; and

20 FIG. 3 is a graphic representation illustrating the delta values for the rib cage and abdominal signals.

25

30

35

40

45

50

55

60

1

-7-

5 Best Mode For Carrying Out The Invention

Referring now to the drawings, apparatus for measuring
respiration volume of the type disclosed in U.S. Patent No.
10 4,306,872 is shown in FIGS. 1 and 2. Two extensible conductive
loops 12, 14 are secured in any suitable fashion to elastic
15 tubes 16, 18, respectively, such that the conductors 12 and 14
extend respectively about the rib cage and abdomen of the subject
10. As the subject 10 breathes, the elastic tubes 16, 18 and
20 conductive loops 12, 14 expand and contract, resulting in
changes in the inductances of the loops. After the inductance
25 of each loop is converted to a proportional signal, the signals
are calibrated and then summed to provide a signal indicative of
tidal volume. Calibration of the signals from the rib cage and
30 abdomen conductors 12 and 14, respectively, is necessary because
the relative contributions of the rib cage and abdomen to tidal
35 volume vary from subject to subject and even in a single subject
with different postures, e.g. standing, supine, etc.

Suitable apparatus for converting the inductances of the
40 conductors 12 and 14 to proportional electrical signals,
calibrating those signals to reflect the proper relative
45 contributions, and then summing those signals to provide a
signal indicative of tidal volume are known to those of ordinary
skill in the art. One such apparatus is disclosed in U.S.
50 patent no. 4,306,872. Another suitable apparatus is marketed by
Nims, Inc., Miami Beach, Florida under the model designation
55 Respigraph TM. Such apparatus are generically illustrated in
block diagram form in FIG. 2, where the blocks 20, 22 represent,
respectively, appropriate circuitry for converting the
60 inductances of the rib cage and abdominal conductors 12, 14 to

1

-8-

5 signals suitable for further processing. The scaling amplifiers
24, 26 represent circuitry for calibrating the signals from the
rib cage and abdomen, respectively, to reflect the relative
10 contributions of the rib cage and abdomen to tidal volume. Once
the scaling amplifiers 24, 26 are properly calibrated, the
15 resulting signals may be summed, as by the summing amplifier 28
in FIG. 2, to yield a signal indicative of tidal volume. The
output signal from the summing amplifier 28 as well as the
20 output signals from the two scaling amplifiers 24 and 26 may
then be displayed as on a graphic recorder 30 or a digital
25 voltmeter 32. Optionally, a microprocessor 34 may be
incorporated in the apparatus for summing the signals from the
scaling amplifiers 24, 26 and/or further processing those
30 signals for diagnostic purposes, all in accordance with
techniques known to those of ordinary skill in the art. If the
35 microprocessor 34 is used to sum the signals from the scaling
amplifiers, the summing amplifier 28 may be eliminated.

The present invention is for an improved method for
40 calibrating the signals from the rib cage and abdomen such that
their sum produces a signal indicative of tidal volume. As is
45 noted above, while various calibration techniques are known to
those of ordinary skill in the art, all the known techniques
possess drawbacks.

50

55

60

5 Turning to the calibration technique of the invention, from Equation A it is known that:

$$V = (K \times RC) + (L \times AB)$$

10

where V is total respiration volume or tidal volume, RC is the rib cage contribution to respiration volume as measured by the rib cage conductor and AB is the abdominal contribution to respiration volume as measured by the abdominal conductor. K and L are calibration factors for the rib cage and abdomen, respectively. Another way of expressing this relationship is:

$$V = Mx[(Z \times RC) + AB] \quad \text{[EQUATION D]}$$

25 where MxZ is equal to K and M is equal to L. It will be apparent that Equation D separates the calibration components into a proportionality factor Z and a scaling factor M. Using this approach, calibration may be viewed as a two step process. The first step is the determination of the correct

35 proportionality factor Z satisfying Equation D, such that

$$V = (Z \times RC) + AB \quad \text{[EQUATION E]}$$

In other words, the proportionality factor Z defines the correct relative contributions of the rib cage (RC) and abdomen (AB) to tidal volume (V). Classically, determination of the proportionality factor Z is determined by an isovolume calibration technique in which the subject breathes against a closed airway i.e. with no volume movement at the mouth, whereby V=0. In other words, during an isovolume maneuver, the only movement of volume is between the rib cage and abdomen compartments, i.e. pendelluft, since no air escapes through the mouth. Under these conditions, Equation D becomes:

$$0 = (Z \times RC) + AB \quad \text{[EQUATION F]}$$

60

or

5 so, by recording the readings from the rib cage and abdomen
conductors during the isovolume maneuver, the proportionality
factor Z can be determined from Equation G. Once Z is
10 determined, the quantity $(Z \times RC) + AB$ can be calculated for any
point in time from the recorded values of the rib cage and
15 abdomen conductors. Since we know from Equation E that
 $(Z \times RC) + AB$ is always proportional to tidal volume V, a
determination of that quantity provides a valuable diagnostic
20 tool. For example, as those of ordinary skill in the art will
appreciate, from this quantity obstructive and central apneas
25 can be diagnosed, RC as a percent of tidal volume V can be
calculated, and increases and decreases in relative tidal volume
V can be assessed.

30 The difficulty with this approach is that an isovolume
maneuver requires breathing against a closed airway, which is not
35 always practical, as in neo-natal and critical care
applications.

40

45

50

55

60

5 If, on the other hand, the proportionality factor Z and hence
the quantity $(Z \times RC) + AB$ is to be determined without an isovolume
manuver, it would appear from Equation E that a measurement of
10 tidal volume must be taken, otherwise there will be a single
equation with two unknowns, namely, proportionality factor Z and
15 tidal volume V .

In accordance with the present invention, this problem is
solved as follows. The rib cage (RC) and abdomen (AB) signals
20 are recorded for a large number of breaths during an initial
baseline period. For example, 250 breaths may be measured
25 during quiet breathing over a 10 minute interval. Actually,
breathing during the baseline period need not be quiet, as long
as it is steady state. For example, the baseline could be
30 derived from breaths recorded during 10 minutes of exercise.
During this baseline period, the uncalibrated signals from the
35 rib cage (RC) and abdomen (AB) conductors are recorded.

Referring to FIG. 3, for each of these signals, two values or
breath "deltas" are calculated for each breath, (Δ_1) being
40 the difference between the signal at the beginning and end of
inspiration, the other (Δ_2) being the difference between the
45 beginning and end of expiration. These delta values are then
totaled separately for each signal. Assuming 250 breaths during
baseline, and since there are two delta values for each breath,
50 the total for each signal will be computed by adding 500 delta
values. Where the delta values described above are proffered,
55 it should be appreciated that the delta values are employed to
provide a parameter indicative of the relative amplitude for
each breath of the uncalibrated rib cage and abdomen signals
60 taken during baseline. Accordingly, as used herein, the term
delta values means any parameter of the uncalibrated rib cage
65 and abdomen signals which provides this information.

5 If actual tidal volume were also recorded during baseline,
as by spirometry, and the deltas for tidal volume totaled and
the mean determined, the following relationship would apply:

10
$$\text{Mean SP} = \text{Mean RC} + \text{Mean AB} \quad [\text{EQUATION H}]$$

Where SP is actual tidal volume as determined, e.g. by
15 spirometry. Since the mean values in Equation H are derived
from uncalibrated signals, calibration factors are required if
the lefthand side in Equation H is to equal the righthand side.
20

The standard deviations (SD) of the mean values for the
25 tidal volume (SP) rib cage (RC) and abdomen (AB) signals can be
calculated. Herein, these will be expressed, respectively, as
SD (SP), SD (RC), and SD (AB). If tidal volume (SP) is constant
30 for all breaths recorded during baseline, Equation H can still
be computed and the standard deviation of tidal volume, SD (V),
35 is 0.

This situation is analogous to Equation F which, as
explained above, applies to the isovolume maneuver where $V=0$. In
40 particular, by considering the standard deviation of a constant
tidal volume SD (V), which is also 0, the pendulluft occurring
45 during normal breathing creates a situation analogous to the
isovolume situation of Equation F, and taking the variance (Var)
of both sides, yields:

50
$$\text{Var} (Z \times \text{RC}) = \text{Var} (\text{AB}) \quad [\text{EQUATION J}]$$

Where variance (Var) is equal to $(\text{SD})^2$. Equation J may be
55 expressed as

$$Z^2 \times \text{Var} (\text{RC}) = \text{Var} (\text{AB}) \quad [\text{EQUATION K}]$$

By taking the square root of both sides

60
$$Z \times \text{SD} (\text{RC}) = \text{SD} (\text{AB}) \quad [\text{EQUATION L}]$$

Which yields

65
$$Z = \text{SD} (\text{AB}) / \text{SD} (\text{RC}) \quad [\text{EQUATION M}]$$

1

-13-

5 It should be appreciated that the standard deviations of the
means of the rib cage and abdomen signals is indicative of the
average variability of those signals. Accordingly, any
10 analysis that provides an indication of average variability of
those signals may be used in lieu of computing standard
15 deviations.

20

25

30

35

40

45

50

55

60

5 put into other words, it will be apparent from Equation M that
the proportionality constant Z for solution of Equation E can be
calculated, assuming constant tidal volume breathing during
10 baseline, from the ratio of the standard deviations of AB and RC
in a manner analogous to the isovolume maneuver, but without the
15 requirement that the airway be blocked.

Of course, taking spirometry readings with a spirometer
during baseline defeats an objective of the calibration
20 technique in accordance with the invention, namely, to calibrate
non-invasively.

25 However, baseline constant tidal volume assumption required to
practice the calibration technique of the invention without
invasively recording actual tidal volumes can be satisfied by
30 removing wild points from the 500 delta values of AB and RC
computed during baseline as by eliminating values outside of 1.5
35 standard deviations of the uncalibrated sum of the RC and AB
components. With those delta values excluded, the remaining
delta values for RC and AB are separately totaled, the means
40 determined, and the standard deviations for the means
calculated. The proportionality factor Z can then be calculated
45 from Equation M.

Once the proportionality factor Z is known, the quantity
(ZxRC)+ AB will always be proportional to actual tidal volume.
50 See Equation E. This quantity can be continuously monitored on
a real time basis from the real time rib cage and abdomen
55 signals generated by the apparatus of FIG. 2. Preferably, this
quantity is expressed as a percent of (ZxMean RC)+Mean AB where
Mean RC and Mean AB are, respectively, the means of the totals
60 of the uncalibrated RC and AB delta values generated at
baseline, but excluding the wild points. This is sufficient for
65 the bulk of diagnostic work, such as detecting obstructive and

1

-15-

5 central apneas, hypoapneas, and variations in tidal volume, the
latter being important as a diagnostic tool in a wide variety of
10 disorders.

15

20

25

30

35

40

45

50

55

60

Referring to FIG. 2, assuming the scaling amplifiers have initially been set to unity gain, once the proportionality factor Z is determined, the scaling amplifier 24 for the rib cage is adjusted to Z. From Equation E, we then know that the sum of the signals from the scaling amplifiers at the output of the summing amplifier 28 will be proportional to tidal volume. This completes the calibration procedure.

While for most purposes a quantitative determination of tidal volume is not necessary, quantitative calibration can be achieved once the proportionality factor Z is known. In particular, referring to Equation D, by taking a single measurement of actual tidal volume V, the scaling factor M can be calculated, since ZxRC and AB are available, respectively, at the output of the rib cage and abdomen scaling amplifiers. In other words, there is then only a single unknown in Equation D, the scaling factor M, which is calculated as:

$$M = V / [(Z \times RC) + AB]. \quad \text{[EQUATION N]}$$

One simple way to take a measurement of actual tidal volume is to simply have the subject inhale a known quantity of air, as from a syringe. Before the subject inhales, this quantity is input to the apparatus of FIG. 2 at the "spirometer", input. The microprocessor 34 can then perform the calculation of Equation N from the values of Z, RC and AB at the end of inspiration from the syringe.

The scaling factor M may be set in the FIG. 2 apparatus by multiplying the gain of the summing amplifier 28 by scaling factor M, whereupon the output of the summing amplifier will be a semi-quantitative indication of tidal volume. The term semi-quantitative is used because it has been determined that tidal volume computed in this fashion is $\pm 10\%$ of actual tidal volume as

1

-17-

5 determined by spirometry.

Desirably, the accuracy of the proportionality factor Z is
10 continuously monitored by repeatedly recalculating the
proportionality factor at five minute intervals and printing out
the resulting value. After the initial calibration procedure,
15 the recalculated value for the proportionality factor Z during
each subsequent five minute interval should be 1.0. In the
event of a change in the position of the patient or other
20 condition that varies the proportionality factor Z , that will
exhibit itself as a reading of Z above or below 1.0. If the
25 variations are too large, i.e. if Z is less than about .7 or
greater than about 1.3, the calibration routine can be rerun.
Suitable visual or audio alarms can be incorporated in the
30 apparatus of FIG. 2 to indicate such excessive variations.

35

40

45

50

55

60

1

-18-

5 As noted above, the calibration technique in accordance with
the present invention is currently utilized in connection with
10 an apparatus for monitoring respiration sold by Nims, Inc.,
Miami Beach, Florida, under their model designation Respigraph
TM. The Respigraph incorporates a microprocessor.
15 Preferably, the microprocessor is programmed to carry out the
calibration technique in accordance with the present invention.
20 A program listing for a suitable program for carrying out the
calibration function appears below in Table A. The program is
in assembly language for a Z80 microprocessor as manufactured by
25 Zylog, Inc.

30

35

40

45

50

55

60

FILE: QDC.ASM

```

;
;
.IDENT QDC
.INSERT SP80.ASM
.INSERT FPMAC.SRC
.INSERT CALCOM.ASM
.INSERT CKCOM.ASM
.INSERT UTCOM
.INSERT SELCOM.ASM
.INSERT VARCOM.CCC
.INSERT GLCOM
.INSERT SYCOM
.INSERT PKCOM
;
.INSERT EFFCOM
.INSERT SIGCOM
.INSERT HRDCOM
.EXTERN VALCOL,VALFND,TVON,SAVECL,LPON,LPOFF,TVOFF
.EXTERN FIND
.EXTERN DERIV5,PRTIME
.EXTERN SAVECL
;
.EXTERN VAL
;
;
.ENTRY QDC1,QDC2,LQDC1
.ENTRY BASE
;
.EXTERN RTON,RTOF,RTICK
;
.EXTERN QDCAL,SYST,ANSOUT
;
;
.EXTERN SRTINIT,SRTPLT,EXEC
.EXTERN LABEL
.EXTERN TDHE
.EXTERN PTRUPD,GETP2,ROUND
;
;
FINIT
;
;
QDC:
;
;THESE JUMPS MUST BE THE 1ST THINGS:
JMP QDC1
JMP QDC2
JMP VAL
JMP LQDC1
JMP ISQV1
JMP BASE

```

-20-

FILE: QDC.ASM

```
;
;
;
;
ISOVI:
    XRA    A
    STA    BSEFLG
    MVI    A,0FFH
    STA    ISOFLG
    MVI    A,0
    STA    WTFLG
    JMP    QDCCOL

;
;
;
;
BASE:
    MVI    A,0FFH
    STA    BSEFLG
    MVI    A,0FFH
    STA    WTFLG
    XRA    A
    STA    ISOFLG
    JMP    QDC3COL

;
;
;
QDC1:
    XRA    A
    STA    BSEFLG
    MVI    A,0
    STA    WTFLG
    STA    ISOFLG
    JMP    QDCCOL

;
LQDC1:
    XRA    A
    STA    BSEFLG
    MVI    A,0FFH
    STA    WTFLG
    XRA    A
    STA    ISOFLG
    JMP    QDCCOL

;
QDCCOL:
;
    XRA    A
    CMA
    STA    QUALFLG ; QUALITATIVE
    XRA    A
    STA    BSEFLG
;
;
```

FILE: QDC.ASM

```

;
QDC3COL:
;
;
;START THE DATA COLLECTION
;
;
XRA    A
STA    VALFLG
STA    POS1    ;NOT IN SINGLE POSITION
LXI    H,0
SHLD   NBERS    ;# VALUES IN XVALUES & YVALUES
;
LXI    H,0
SHLD   TIME
LXI    H,0
SHLD   MAXCNT
;
LDA     BSEFLG
IF (.A,IS,ZERO) ;NOT COLLECTING BASELINE
;INIT THE TRANSDUCESERS
MVI     A,10
CALL    SYST
;THIS RESETS CALS TO 1 AND 1 SO AFTER THIS IS EXECUTED, ALL OTHER
; CALIBRATIONS ARE NULLIFIED
XRA     A
STA     CALTYPE
STA     LASTCAL ;RIGHT NOW UNCALBRATED
STA     GOODBASE    ;NO LONGER HAS GOOD BASELINE
;
CALL    LPON
FPRN    RES,RESPT
CALL    LPOFF
ENDIF
;
CALL    SETUP
CALL    SRTINIT ;INIT RT. PLOT
LOOP
CALL    SRTPLT
;
LDA     BFLG
IF (.A,IS,NZERO)
;UPDATE # DELTAS AND ELAPSED TIME
MVI     A,3
STA     OPCODE
CALL    ANSOUT
ELSE
LDA     SECFLG
IF (.A,IS,NZERO)
MVI     A,3
STA     OPCODE

```

-22-

FILE: QDC.ASM

```

CALL ANSOUT
LDA  FLSHFLG
DCR  A
STA  FLSHFLG
IF (.A,IS,ZERO)
    LXI  H,TOPLFT+53
    LDA  BSEFLG
    IF (.A,IS,ZERO)
        LDA  ISOFLG
        IF (.A,IS,ZERO)
            LXI  D,CLBMESS
        ELSE
            LXI  D,ISOMESS
        ENDIF
    ELSE
        LXI  D,BSEMESS
    ENDIF
    CALL  LABEL
    MVI  A,2
    STA  FLSHFLG
ELSE
    LXI  H,TOPLFT+53
    LXI  D,BLANKS
    CALL  LABEL
ENDIF

E:
ENDIF

;
;
; LDA  GFLG      ;= OFFH THEN CLOCK
;                      ;STILL ON
; EXITIF(.A,IS,ZERO)
; ENDLLOOP
;
;
; NOW DATA COLLECTION PHASE HAS BEEN COMPLETED
; MAXCNT HOLDS THE NUMBER OF MAX POINTS COLLECTED
;
; IRSRV USES CHANNEL SPBUFF FOR DETECTIONS
; SPMAX HOLDS THE DELTAS FROM SPBUFF
; RCMAX HOLDS THE DELTAS FROM RCBUFF
; ABMAX HOLDS THE DELTAS FROM ABBUFF
;
; CLEAR THE SCREEN
CALL  TVON
CALL  TVOFF
;
;
LDA  BSEFLG
IF (.A,IS,ZERO)
LDA  ISOFLG
IF (.A,IS,NZERO)
JMP  ISOCALC
;THIS WAS AN ISOVOLUME MANUEVER

```

-23-

FILE: QDC.ASM

```

ENDIF
ENDIF
;
LHLD    MAXCNT
LXI     0,3
ORA     A
DSBC    0
IF (PSW,IS,CARRY)
;
    LDA    WFLG
    IF (.A,IS,ZERO)
        MVI A,1
        STA OPCODE ;NOT ENOUGH DELTAS
        CALL    ANSOUT
        MVI A,0
        STA OPCODE ;PRINT MENU ON SCREEN
        CALL    ANSOUT
    ENDIF
    STC
    RET
ENDIF
;
;
;
;THERE ARE 2*(MAXCNT -1) FLOATING PT VALUES IN XVALUES
; AND YVALUES FOR THIS RUN
    LHLD    MAXCNT
    SHLD    NBERS ;TOTAL # STORED
    CALL    QDCAL
;RETURN WITH CARRY SET IF BAD FOR ANY REASON
;
;
;ORA    A
;MVI    A,0
;STA    OPCODE ;PRINT MENU ON SCREEN
;CALL    ANSOUT
;RET
;
;
;
;
;SOCALC:
    LDA     WRAPF
    IF (.A,IS,NZERO) ;THEN 400 PTS AND RIPT PTS TO 1ST
                        ;DATA PT
        LXI     H,400
        SHLD    CNT
        LHLD    RIPT
        SHLD    ROPT
    ELSE
;TIME HOLDS # OF DATA POINTS AND 1ST PT AT OFFSET = 0
        LHLD    TIME
        SHLD    CNT

```

-24-

FILE: GDC.ASM

```

        LXI    H,0
        SHLD   ROPT
    ENDIF
;
    FFOA    0
    FSTA     XSUM
    FSTA     YSUM
    FSTA     XXSUM
    FSTA     XYSUM
    FSTA     XCNT
LOOP1:
    CALL     GETX
    CALL     GETY
    CALL     XYPAIR
;
    LHLD     ROPT
    LXI      B,RSIZE
    CALL     PTRUPD
    SHLD     ROPT
    DSKZ     CNT,LOOP1
;
    CALL     LSTFIT
    FLDA     SLOPE
    FABS
    FSTA     ABCAL
    FFOA     1
    FSTA     RCCAL
;PRINT IT ON SCREEN
    MVI      A,4
    STA      OPCODE
    CALL     ANSOUT
;
    CALL     SAVECL
    IF (PSW,IS,NCARRY) ;THEN GOOD VALUES
        MVI  A,3 ;ISOV QUAL
        STA  CALTYPE
        MVI  A,0FFH
        STA  LASTCAL ;GOOD CALIBRATION
        XRA  A
        CMA
        STA  QUALFLG
        CALL LPON
        CALL PRTIME
        FPRN ISV,ISVPT
        CALL LPOFF
        ORA  A
    ENDIF
;
    RET
;
;
;

```

-25-

FILE: QDC.ASM

```
;
ISV:  .ASCIS
      QUALITATIVE ISOVOLUME CALIBRATION
      RCCAL = 303
      ABCAL = 303
;
ISVPT: .WORD  RCCAL,ABCAL
;
;
GETX:
      LXI      H,ABBUFF
      LDED     ROPT
      DAD      D
      MOV      E,M
      INX      H
      MOV      D,M
      SDED     XVAL
      FILA     XVAL
      FLOT
      FSTA     XVAL
      RET
;
GETY:
      LXI      H,RCBUFF
      LDED     ROPT
      DAD      D
      MOV      E,M
      INX      H
      MOV      D,M
      SDED     YVAL
      FILA     YVAL
      FLOT
      FSTA     YVAL
      RET
;
XYPAIR:
      FLDA     XVAL
      FLDB     XSUM
      FADD
      FSTA     XSUM
;
      FLDB     YVAL
      FLDA     YSUM
      FADD
      FSTA     YSUM
;
      FLDB     XVAL
      FLDA     XVAL
      FMUL
      FLDB     XXSUM
      FADD
      FSTA     XXSUM
;
```

FILE: QDC.ASM

```

        FLDB    XVAL
        FLDA    YVAL
        FMUL
        FLDB    XYSUM
        FADD
        FSTA    XYSUM
;
        FLDB    XCNT
        FFOA    1
        FADD
        FSTA    XCNT
        RET

```

LSTFIT:

```

        FLDA    XSUM
        FLDB    XSUM
        FMUL
        FPSH
;
        FLDA    XCNT
        FLDB    XXSUM
        FMUL
        FSWP
        FPOP
;
        FSUB
        FPSH
;
        FLDA    XSUM
        FLDB    YSUM
        FMUL
        FPSH
;
        FLDA    XCNT
        FLDB    XYSUM
        FMUL
        FSWP
        FPOP
;
        FSUB
        FSWP
        FPOP
        FDIV
        FSTA    SLOPE
        RET

```

;

QDC2:

```

;THIS CAN BE DONE AFTER EITHER A QDC1 (CALTYPE = 1/2)
; OR AFTER AN ISOVOLUME (CALTYPE = 3/4)
;
;IF CALTYPE = 0, THEN HAVEN'T CALIBRATED THE RC TO AB

```

-27-

FILE: QDC.ASM

```

        LDA    CALTYPE
        IF (.A,IS,ZERO)
            MVI    A,2
            STA    OPCODE
            CALL    ANSOUT
            RET
        ENDIF
;
;NOW COLLECT DATA AS IF VALIDATING
        CALL    VALCOL
;
        LHLD    MAXCNT
        LXI    D,2
        ORA    A
        DSBC    D
        IF (PSW,IS,CARRY)            ;THEN LESS THAN 2 BREATHS
            MVI    A,1
            STA    OPCODE
            CALL    ANSOUT
            MVI    A,0
            STA    OPCODE ;PRINT MENU ON SCREEN
            CALL    ANSOUT
            RET
        ENDIF
;
;NOW FIND THE RATIOS OF SUM/KNOWN VOLUME
        CALL    VALFND
;RETURNS WITH AREG=# OF VALUES USED IN SUMMATION
;
        FSTA    FCNT
;
;
;NOW XVALUES HOLDS ALL OF THE RATIOS, XSUM HOLDS SUM OF ALL XVALUES
;XXSUM HOLDS SUM OF SQUARES OF ALL XVALUES, AND FCNT HOLDS THE NUMBER
;OF VALUES USED IN THE SUMMATION
;
;COMPUTE THE MEAN RATIO
        FLDB    XSUM
        FLDA    FCNT
        FDIU
        FSTA    MEAN
;NOW KNOW THAT SUM NEEDS TO BE MULTIPLIED BY 1/MEAN, SO
;RCCAL = CURRENT RC GAIN (RCGNF) * 1/MEAN
;ABCAL = CURRENT AB GAIN (ABGNF) * 1/MEAN
        FFOA    1
        FATB
        FLDA    MEAN
        FDIU
        FLDB    RCGNF
        FMUL
        FSTA    RCCAL
;
        FFOA    1

```


FILE: QDC.ASM

```
;MVI A,0
;STA OPCODE ;PRINT MENU ON SCREEN
;CALL ANSOUT
RET

;;
;
UL2: .ASCIS /
QUANTITATIVE CAL FACTORS
RC CAL = 3D3
AB CAL = 3D3

;
UL2PT: .WORD RCCAL,ABCAL
;
;
;NOTE: MCA VALUES OF RCINVOL, ABINVOL, AND SINVOL WILL
; BE STORED BEGINNING AT ADDR = XVALUES
; XVALUES + YVALUES = 1600 BYTES
; I WILL STORE UP TO 1500 BYTES
;
; == > 500 DELTAS = 250 INSP DELTAS * 6 BYTES
; = 1500 BYTES
; MCAAADDR = ADDRESS OF NEXT TO STORE
;
;
;
SETUP:
;THIS IS ONLY USED WHEN CALIBRATING QDC METHOD
;SETUP THE CLOCK
;SAMPLE AT 20 PTS PER SEC
PUSH H
; INIT REAL-TIME BUFFER POINTERS
; MODIFIED 09-24-81
;FOR PEAK FLOWS:
LXI H,DLTIMES
SHLD TMPTR
LXI H,0
SHLD TTPOS
SHLD TTNEG
SHLD TEMPK
SHLD TTPTM
SHLD TTNMTM
SHLD PKTIME
;
LXI H,0
SHLD DLTIPTR ;OFFSET TO STORE NEXT DELTA
XRA A
STA DLTFULL ;DELTA RING BUFFER NOT FULL YET
```

FILE: QDC.ASM

```

CMA
STA FSTMN
;
LXI H,XVALUES ;WILL USE THIS BUFFER TO
SHLD MCAADDR ;MCA VALUES
;
LXI H,0
SHLD SPTMP
;
XRA A
STA FRSTMX ;1ST MAX NOT FOUND
;
LXI H,TMBUFF
MVI M,0
INX H
MVI M,0
;
MVI A,2
STA FLSHFLG
;
;
XRA A
STA WRAPF
STA SECFLG
CMA
STA BFLG
LXI H,0
SHLD ELPTM
SHLD ELPMIN
SHLD SECONDS
;
LXI H,0
SHLD RIPT
SHLD ROPT
SHLD ODRIPT
LXI H,0
SHLD TIME
; INIT VARIABLES FOR PICKING MAX AND MINS
XRA A
STA MAXB
STA PTF ;START BY LOOKING FOR MAX PT
STA SLOPEF ;INIT MAX NOT FOUND
;
XRA A
STA INTFLG
CMA
STA GFLG ;INDICATES CLOCK ON
;
LXI H,IRSRV
CALL RTON ;TURN ON CLOCK
;
POP H

```

-31-

FILE: QDC.ASM

```

    RET
;
;
;
CLKSTP:
    ;STOP DATA COLLECTION
    ;TURN CLOCK OFF
;
;MODIFIED 09-24-81
;
    CALL    RTOF    ;TURN OFF CLOCK
    XRA     A
    STA     GFLG
    POP     D
    POP     H
    POP     B
    POP     PSW
    RET
;
;
;
;
;
IRSRV:
;
; INTERRUPT SERVICE ROUTINE
;
    PUSH    PSW
    PUSH    B
    PUSH    H
    PUSH    D
    CALL    RTICK    ;SERVICE INTERRUPT
    LDA     INTFLG
    IF (.A,IS,NZERO)    ;THEN INTERRUPT INTERRUPT
        NOP
    ELSE
        CMA
        STA     INTFLG    ;SET FLAG
    ENDIF
;
;
    LHLD    ELPTM
    INX     H
    SHLD    ELPTM
    LDED    SFREQ
    CALL    TDHE
    IF (PSW,IS,ZERO)    ;THEN 1 SECON
        LXI     H,0
        SHLD    ELPTM
        LHLD    SECONDS
        INX     H
        SHLD    SECONDS
        XRA     A

```

-32-

FILE: QDC.ASM

```

CMA
STA SECFLG
LXI D,60
CALL TDHE
IF (PSW,IS,ZERO) ;THEN 1 MINUTE
    LXI H,0
    SHLD SECONDS
    LHLD ELPMIN
    INX H
    SHLD ELPMIN
;IF COLLECTING BASELINE: THEN STOP AT 5 MINUTES
LDA BSEFLG
IF (.A,IS,NZERO)
    LXI D,5
    LHLD ELPMIN
    CALL TDHE
    IF (PSW,IS,ZERO)
        JMP CLKSTP
    ENDIF
ELSE
;
;
;WHILE CALIBRATING: IF REACH 10 MINUTES BEFORE 500 DELTAS,
; THEN AUTOMATICALLY STOP
    LXI D,10
    LHLD ELPMIN
    CALL TDHE
    IF (PSW,IS,ZERO) ;HAVE REACHED 10 MINUTES
        JMP CLKSTP
    ENDIF
ENDIF
;
ENDIF
ENDIF
;
;
;
;ALWAYS COLLECT RC INTO RCBUFF,
; AB INTO ABBUFF,
;
;NOTE: ALWAYS USES DATA IN SPBUFF FOR DETECTION
;
    LHLD RCAD ;RIB CAGE VALUE
    MOV A,H
    CMA
    ANI 0FH
    MOV H,A
    MOV A,L
    CMA
    MOV L,A
    SHLD DATA
    SHLD RCV

```

-33-

FILE: QDC.ASM

```

;
    LHLD    RIPT
    LXI     D,RCBUFF
    CALL    PRNG
;
;
    LHLD    ABAD      ;ABDOMEN VALUE
    MOV     A,H
    CMA
    ANI     0FH
    MOV     H,A
    MOV     A,L
    CMA
    MOV     L,A
    SHLD    DATA
    SHLD    ABV
;
    LHLD    RIPT
    LXI     D,ABBUFF
    CALL    PRNG
;
; SUM INTO SPBUFF--- TO PICK OFF BREATHS
    LHLD    RCV
    LDED    ABV
    DAD     D
    SHLD    DATA
;
    LHLD    RIPT
    LXI     D,SPBUFF
    CALL    PRNG
;
;
;
;
; INCREMENT RIPT FOR NEXT TIME
;
    LHLD    TIME
    INX     H
    SHLD    TIME
    SHLD    DATA
    LHLD    RIPT
    LXI     D,TMBUFF
    CALL    PRNG
;
    LHLD    RIPT
    SHLD    ODRIPT
;
    LXI     B,RSIZE
    LHLD    RIPT
    CALL    PTRUPD
    SHLD    RIPT

```

-34-

FILE: QDC.ASM

```

;
LDA    WRAPF
IF (.A,IS,ZERO) ;SEE IF RIPT RESET TO 0
    LXI    D,0
    LHLD   RIPT
    CALL   TDHE
    IF (.PSW,IS,ZERO)
        MVI A,OFFH
        STA WRAPF
    ENDIF
ENDIF

;
LDA    BSEFLG
IF (.A,IS,ZERO)
LDA    ISOFLG
IF (.A,IS,NZERO) ;THEN NO PEAK DETECTION
    JMP    EXIT
ENDIF
ENDIF

;
;
LDA    SLOPEF
IF (.A,IS,ZERO) ;THEN INIT POINTS NOT COLLECTED YET
    CALL   GETP2 ;RETURNS C IF < 5 PTS IN BUFFER
                ;THIS JUST MAKES SURE EXTRA
                ;POINTS IN BUFFER
    JC     EXIT ;NOT ENOUGH POINTS IN BUFFER
;NOTE: GETP2 INITIALIZES ROPT TO LINE UP WITH DERIV STUFF
;SKIP 1ST 3 POINTS IN BUFFER
; LXI H,6 ;OFFSET POINTING TO POINT 4
; SHLD    ROPT

;
;
    MVI    A,OFFH
    STA    SLOPEF
ENDIF

;
;
;
CALL    DERIVS

;
;
*****
;MUST ADD THE CODE TO DO PEAK DETECTION
;HERE TOO !!
;
LDA    FRSTMX
IF (.A,IS,ZERO) ;HAVE NOT FOUND 1ST MAX POINT
;

```

-35-

FILE: QDC.ASM

```

;
;
    LHL RPT
    CALL FIND
    IF (PSW,IS,NCARRY) ;NOT A MAX
        JMP OUTER
    ENDIF
;HAVE 1ST MAX
    LDA FRSTMX
    CMA
    STA FRSTMX
    SHLD SPTMP ;TEMP CRITICAL POINT
    SDED RCTMP
    SBCD ABTEMP
    LDA PTF
    CMA
    STA PTF ;NOW LOOK FOR MIN
    JMP OUTER
ENDIF

;
;
    LHL RPT
    CALL FIND
    IF (PSW,IS,CARRY)
        SHLD SPVALUE
        SDED RCVALUE
        SBCD ABVALUE
        LDA PTF
        IF (.A,IS,ZERO) ;THIS IS A MAX POINT
            LHL RCVALUE
            SHLD RCMXVAL
            LHL ABVALUE
            SHLD ABMXVAL
        ;
        CALL DLTSTORE
        ;ONLY STORE THE MCA VARIABLES AT A MAX (ONLY WANT FOR INSP)
        CALL MCASTORE
        ;
        ;
        ;IF DLTIPTR = 1000, THEN EXIT
        LHL DLTIPTR
        LXI D,1000
        CALL TDHE
        IF (PSW,IS,ZERO)
            MVI A,OFFH
            STA DLTFULL
            LXI H,0
            SHLD DLTIPTR
            JMP CLKSTP
        ENDIF
    ELSE ;THIS IS MIN POINT
        LHL RCVALUE
        SHLD RCMVAL

```

-36-

FILE: QDC.ASM

```

        LHL D    ABVALUE
        SHL D    ABMVAL
        LDA      FSTMN
        IF (.A,IS,NZERO)      ;THIS IS THE 1ST MIN, DO NOT STORE
            LDA      FSTMN
            CMA
            STA      FSTMN
        ELSE      ;STORE THE DELTA
            CALL     DLTSTORE
        ;
        ;
        ;IF DLTIPTR = 1000, THEN EXIT
        LHL DLTIPTR
        LXI D,1000
        CALL TDHE
        IF (PSW,IS,ZERO)
            MVI A,OFFH
            STA DLTFULL
            LXI H,0
            SHL DLTIPTR
            JMP CLKSTP
        ENDIF
    ENDIF
;
    ENDIF
    LDA PTF
    CMA
    STA PTF
ENDIF
;
;*****
OUTER:
;
;UPDATE ROPT FOR NEXT TIME
;
    LHL ROPT
    LXI B,RSIZE
    CALL PTRUPD
    SHL ROPT
;
EXIT:
; LOOK FOR ENTER
;
    CALL CSTS
    IF (PSW,IS,NZERO)
        CALL CINP
        IF (.A,EQ,00H)      ;IF ENTER
            JMP CLKSTP
        ENDIF
    ENDIF
;
    XRA A

```

-37-

FILE: QDC.ASM

```

        STA    INTFLG
        POP    D
        POP    H
        POP    B
        POP    PSW
        RET

;
;
MCASTORE:
        LHL    MCAAADDR
        LHLD   RCINVOL
        MOV    M,E
        INX    H
        MOV    M,D
        INX    H
        LHLD   ABINVOL
        MOV    M,E
        INX    H
        MOV    M,D
        INX    H
        LHLD   SMVOL
        MOV    M,E
        INX    H
        MOV    M,D
        INX    H
        SHLD   MCAAADDR
        RET

;
;
;
DLTSTORE:
        LHL    RCMXVAL
        LHLD   RCMNVAL
        ORA    A
        DSBC   D
;HL = INSP VOLUME      ;RC
        SHLD   RCVALUE ;DELTA RC
        LHL    DLTIPTR ;OFFSET TO STORE
        LXI    D,RCDATA
        DAD    D
        LHLD   RCVALUE
        MOV    M,E
        INX    H
        MOV    M,D

;
        LHL    ABMXVAL
        LHLD   ABMNVAL
        ORA    A
        DSBC   D
;HL = INSP VOLUME      ;AB
        SHLD   ABVALUE ;DETLA AB
        LHL    DLTIPTR ;OFFSET TO STORE
        LXI    D,ABDATA

```

FILE: QDC.ASM

```

        DAD      D
        LDED     ABVALUE
        MOV      M,E
        INX      H
        MOV      M,D
        LHLD     DLTIPTR
        INX      H
        INX      H
        SHLD     DLTIPTR
        LHLD     MAXCNT
        INX      H
        SHLD     MAXCNT
        XRA      A
        CMA
        STA      BFLG
;STORE THE TIMES
        LHLD     TMPTR
        LXI      D,DLTEND
        CALL     TDHE
        IF (PSW,IS,ZERO)      ;NO ROOM
            RET
        ENDIF
;
        LHLD     TMPTR
        LDED     OPTIME ;TIME OF THE CP
        MOV      M,E
        INX      H
        MOV      M,D
        INX      H
        LDED     PKVALUE ;TIME OF THE PEAK FLOW
        MOV      M,E
        INX      H
        MOV      M,D
        INX      H
        SHLD     TMPTR ;FOR NEXT TIME
        RET
;
;
PRNG:
;
;HL HOLDS THE RING INPUT PTR
;DE HOLDS ADDR OF RING
;
        DAD      D
        LDED     DATA
        MOV      M,E
        INX      H
        MOV      M,D
        RET
;
;ISOMESS:      .ASCIS 'Isovolume'
;
;BSEMESS:      .ASCIS 'Baseline'

```

-39-

FILE: QDC.ASM

```

;
CLBMESS:      .ASCIS  'Calibrating'
BLANKS:       .ASCIS
;
RES:          .ASCIS  '
AT 31:31:31   Cal factors reset to 1 and 1
;
RESPT: .WORD   HOUR,MIN,SEC
;
          .EXTERN QDCOM
          .LOC    QDCOM
;
MAXB:  .BLKB   1
RCCU:  .BLKB   2
ABCU:  .BLKB   2
SPCU:  .BLKB   2
RCV:   .BLKB   2
ABV:   .BLKB   2
MEAN:  .BLKB   4
FCNT:  .BLKB   4
;
FRSTMX:      .BLKB   1
RCMXVAL:     .BLKB   2
RCMIVAL:     .BLKB   2
ABMXVAL:     .BLKB   2
ABMIVAL:     .BLKB   2
FSTMN: .BLKB   1
FLSHFLG:     .BLKB   1
ISOFLG:      .BLKB   1
XCNT:  .BLKB   4
QDCXTRA:     .BLKB   9      ;EXTRA LOCS
          .RELOC
          .END    QDC

```

```

;
;
;
;

```

FILE: QDCAL.ASM

40

```

        .IDENT  QDCAL
        .INSERT FPMAC.SRC
        .INSERT SP80
        FINIT
        .INSERT VTCCOM
        .INSERT CALCOM
        .INSERT OLQCOM
        .INSERT CKCOM
        .INSERT VARCOM.CCC
        .INSERT SYCOM
        .INSERT EFFCOM
        .INSERT SIOCOM
        .INSERT QD1.ASM
        .INSERT PKCOM

;
        .EXTERN CMPSTATS
;
        .ENTRY  QDCAL
        .EXTERN TDHE,ROUND
        .EXTERN SAVECL,ANSOUT,LPON,LPOFF
        .ENTRY  V2      ;SO CAN LOOK AT MAP!
        .EXTERN PRTIME
        .EXTERN WTSAVECL
;
        .EXTERN TVON,TVOFF
;
;
;
QDCAL:
        LDA     BSEFLG
        IF (.A,IS,NZERO)      ;BASELINE
            JMP  BCOMP
        ENDIF

;ENTERED HERE. THEN IN CALIBRATE MODE (NOT LONG TERM)
;
        CALL    CMPSTATS
        RC      ;CARRY SET IF BAD CAL
                ;BECAUSE NOT ENOUGH DATA COLLECTED
;
;
;ABCAL = RCSTD / ABSTD
        FLDB    RCSTD
        FLDA    ABSTD
        FDIV
        FSTA    ABCAL
;RCCAL = 1
        FFOA    1
        FSTA    RCCAL
;
;
;MUST STORE THE GAINS, BECAUSE RESCALE ASSUMES SO !!
        LDA     WTF LG

```

FILE: QDCAL.ASM

```

IF (.A,IS,ZERO) ;WANT TO PRINT ERROR CODE IF BAD
    CALL SAVECL
ELSE
    CALL WTSAVECL
ENDIF
IF (PSW,IS,CARRY) ;THEN BAD CALS
    MVI A,0 ;NO GOOD CALIBRATION (WHEN BEGAN THIS RUN, RESET
    STA CALTYPE ;CALS TO 1 AND 1
    MVI A,0
    STA LASTCAL ;NO GOOD CAL
    STA GOODBASE ;NO GOOD BASELINE NOW
;PRINT OUT WHAT THE BAD CALS WERE AND THE STDS
    CALL BDCALS
    STC ;BAD CALS
    RET
ENDIF

;
;
;NOW MUST USE THE MEAN VT MEASURED TO SCALE THE CAL FACTORS S8
; MEAN VT = VTMX/10
    CALL RESCALE

;
LDA WFLG
IF (.A,IS,ZERO)
    MVI A,4
    STA OPCODE
    CALL ANSOUT ;DISPLAY THE CALS ON THE MONITOR
ENDIF

;
LDA WFLG
IF (.A,IS,ZERO) ;WANT TO PRINT ERROR CODE IF BAD
    CALL SAVECL
ELSE
    CALL WTSAVECL
ENDIF
IF (PSW,IS,NCARRY) ;THEN GOOD CALS
    CALL LPON
    CALL PRTIME
    FPRN V1,V1PT
    CALL LPOFF
    CALL CMPMCA ;COMPUTE MCA/VT
    CALL LPON
    FPRN V2,V2PT
    CALL PRTHYP
    CALL LPOFF
    MVI A,1 ;QDC QUAL
    STA CALTYPE
    MVI A,0FFH
    STA LASTCAL
    ORA A ;GOOD CALS
ELSE
    MVI A,0 ;NO GOOD CALIBRATION (WHEN BEGAN THIS RUN, RESET
    STA CALTYPE ;CALS TO 1 AND 1

```

FILE: QDCAL.ASM

```

      MVI    A,0
      STA    LASTCAL ;NO GOOD CAL
      STA    GOODBASE ;NO GOOD BASELINE NOW
;PRINT OUT WHAT THE BAD CALS WERE AND THE STDS

```

```
CALL BDCALS
FFOA 1
FSTA RCCAL
FSTA ABCAL
CALL CMPMCA ;COMPUTE MCA/VT
CALL L9ON
FPRN U2,U2PT
CALL PRTHYP
LXI H,STAR3
CALL TXTYP
CALL LPOFF
```

```

      STC          ;RAD CALS
ENDIF
RET

```

SDCALS:

```
CALL LPRN
LXI H,STARS
CALL TXTYP
CALL PRTIME
FPRN BADC,BAOPT
CALL LPOFF
RET
```

PRTHYP:

```

FPRN      HYP,HYPPT
LDA        QUALFLG
IF (.A,IS,NZERO)      ;QUAL
LXI H,RHYP
ELSE      ;QUANT
LXI H,AHYP
ENDIF
CALL      TXTYP
RET

```

BCOMP:

```

;THIS COMPUTE MEAN VT AND IF ISDV WAS QUALITATIVE, CALTYPE = 3,
; THEN IT SCALES IT TO VTMAX/10

```

11

-43-

FILE: GDCAL.ASM

```

CALL    CMPSTATS
IF (PSW,IS,ZERO)      ;NOT ENOUGH DATA
    XRA    A
    STA    GOODBASE    ;NOT A GOOD BASELINE
    STC
    RET
ENDIF

;
CALL    RESCALE ;THIS WILL CHANGE VALUE OF RCCAL AND ABCAL
;
LDA     CALTYPE
IF (A,EQ,3)      ;ISO QUAL
;
;SAVE CALS
;NOW MUST USE THE MEAN VT MEASURED TO SCALE THE CAL FACTORS SO
;MEAN VT = VTMX/10
CALL    WTSW-ECL
;
;
IF (PSW,IS,NCARRY)    ;THEN GOOD CALS
    MVI    A,5      ;ISO-QUAL WITH BASELINE
    STA    CALTYPE
    MVI    A,OFFH
    STA    LASTCAL ;GOOD CAL
    CALL    LPON
    CALL    PRTIME
    FPRN    BS1,BS1PT
    CALL    LPOFF
    CALL    CMPCMCA ;COMPUTE MCA/VT
    CALL    LPON
    FPRN    V2,V2PT
    CALL    PRTHYP
    CALL    LPOFF
    MVI    A,OFFH
    STA    GOODBASE    ;GOOD BASELINE
    XRA    A          ;GOOD CALS
ELSE
    FFOA    1
    FSTA    RCCAL
    FSTA    ABCAL
    CALL    CMPCMCA ;COMPUTE MCA/VT
    CALL    LPON
    LXI    H,STARS
    CALL    TXTYP
    CALL    PRTIME
    FPRN    BSBAD,BSBPT
    FPRN    V2,V2PT
    CALL    PRTHYP
    LXI    H,STARS
    CALL    TXTYP
    CALL    LPOFF
    MVI    A,0
    STA    GOODBASE    ;BAD BASELINE

```

-44-

FILE: QDCAL.ASM

```

        STC                ;BAD CALS
    ENDIF
ELSE    ;ALREADY DID QUANT - 190V
    IF (.A.EQ.5)
        MVI A,5           ;150 QUANT WITH BASELINE
        STA CALTYPE
        CALL LPON
        CALL PRTIME
        FPRN 853.853PT
        CALL LPOFF
;CALS DO NOT CHANGE
        FFOA 1
        FSTA RCCAL
        FSTA ABCAL
        CALL CMPMCA ;COMPUTE MCA/VT
        CALL LPON
        FPRN V2,V2PT
        CALL PRTHYP
        CALL LPOFF
        MVI A,0FFH
        STA GOODBASE      ;GOOD BASELINE
        ORA A              ;GOOD CALS
    ELSE
        IF (.A.EQ.7)      ;TYPED IN CALS, QUANT, JUST GETTING BASELINE STUFF
            LDA QUALFLG
            IF (.A.IS,2500)
                CALL LPON
                CALL PRTIME
                FPRN 857.857PT
                CALL LPOFF
;CALS DO NOT CHANGE
                FFOA 1
                FSTA RCCAL
                FSTA ABCAL
                CALL CMPMCA ;COMPUTE MCA/VT
                CALL LPON
                FPRN V2,V2PT
                CALL PRTHYP
                CALL LPOFF
                MVI A,0FFH
                STA GOODBASE ;GOOD BASELINE
                ORA A        ;GOOD CALS
            ELSE
                ;TYPED IN CALS, QUAL SO WILL CHANGE CALS
            **
            ;
            ;SAVE CALS
            ;NOW MUST USE THE MEAN VT MEASURED TO SCALE THE CAL FACTORS SO
            ; MEAN VT = UTMX/10
            CALL WTSAVECL
            ;
            ;
            IF (PSW,IS,NCARRY) ;THEN GOOD CALS
                CALL LPON

```

FILE: QDCAL.ASM

```

        CALL      PRTIME
        FPRN      V7,V7PT
        CALL      LPOFF
        CALL      CMPCMCA ;COMPUTE MCA/VT
        CALL      LPON
        FPRN      V2,V2PT
    CALL PRTHYP
    CALL      LPOFF
    MVI A,0FFH
    STA GOODBASE ;GOOD BASELINE
    ORA A      ;GOOD CALS
    ELSE
        FFOA      1
        FSTA      RCCAL
        FSTA      ABCAL
        CALL      CMPCMCA ;COMPUTE MCA/VT
        CALL      LPON
        CALL      PRTIME
        FPRN      BSBAD,BSBPT
        FPRN      V2,V2PT
    CALL PRTHYP
    CALL      LPOFF
    MVI A,0
    STA GOODBASE ;BAD BASELINE
    STC          ;BAD CALS
    ENDIF
; **
    ENDIF
    ENDIF
    ENDIF

    ENDIF
    RET

;
;
;
CMPCMCA:
;
;
; COMPUTE MEAN MCA/VT
; MUST REMEMBER THAT ALL RC VALUES MUST BE MULT BY RCCAL
; AND ALL AB VALUES MUST BE MULT BY ABCAL
;
; NOTE: WILL NOT BE ABLE TO COMPUTE MCA/VT THE CLEAN WAY OF USING
; THEN SUM OF THE SUM'S DERIV BECAUSE THE THE CAL FACTOR PROBLEM
; SO WILL HAVE TO DIVIDE BY RC DELTA * RCCAL + AB DELTA * ABCAL
    LXI      H,XVALUES
    SHLD     ADDR1
    LDA      DLTFULL
    IF (.A,IS,NZERO) ;FULL
        LHL DLTIPTR
        SHLD OFFSET

```

```

ELSE
    LXI    H,0
    SHLD   OFFSET
ENDIF

FF0A     0
FSTA     VTSUM
FSTA     MCASUM
FSTA     XCNT
LOOP

LHLD     ADDR1
LDOD     MCAADDR
CALL     TOHE
IF (PSW,IS,ZERO)
    RET
ENDIF

;
;GET MCA VALUE
LHLD     ADDR1
MOV      E,M
INX      H
MOV      D,M
SDOD     RCMCA
INX      H
MOV      E,M
INX      H
MOV      D,M
SDOD     ABMCA
INX      H
MOV      E,M
INX      H
MOV      D,M
SDOD     SMMCA
INX      H
SHLD     ADDR1

;
FILE     RCMCA
FLOT
FLDB     RCCAL
FMUL
FPSH
FILE     ABMCA
FLOT
FLDB     ABCAL
FMUL
FATB
FPOP
FADD
FATB
FF0A     15      ;BECAUSE OF 9 PT DERIV 60
              ;/4 BECAUSE OF DIVISION WHEN COLLECTED
FDIV
FLDB     CCS

```

-47-

FILE: QDCAL.ASM

```

        FMUL
        FSTA      MCAVALUE
;
;GET VT VALUE
        LHL0      OFFSET
        LXI        D,RC0DATA
        DAD        D
        MOV        E,M
        INX        H
        MOV        D,M
        SDE0      FRC
;
        LHL0      OFFSET
        LXI        D,AB0DATA
        DAD        D
        MOV        E,M
        INX        H
        MOV        D,M
        SDE0      FAB
;
        FILA      FRC
        FLOT
        FLOB      RCCAL
        FMUL
        FPSH
        FILA      FAB
        FLOT
        FLOB      ABCAL
        FMUL
        FATB
        FPOP
        FADD
        FLDB      CCS
        FMUL
        FSTA      VTVALUE
;
;
;MCASUM = SUM OF MCAVALUE,VTVALUE
;
        FLDB      MCAVALUE
        FLDA      VTVALUE
        FDIV
;IF < 1, THEN MAKE = 1
        FPSH
        FATB
        FFOA      1
        FSUB
        FTST      AREG,LT
        IF (PSW.IS,ZERO)
            FPOP
            FFOA      1
        ELSE
            FPOP

```

FILE: QDCAL.ASM

```

        ENDIF
;
        FLDB    MCASUM
        FADD
        FSTA    MCASUM
;
        FLDB    VTVALUE
        FLDA    VTSUM
        FADD
        FSTA    VTSUM

        FLDB    XCNT
        FFOA    1
        FADD
        FSTA    XCNT
;
        LHLD    OFFSET
        INX     H
        INX     H
        SHLD    OFFSET
        LDLD    DLTIPTR
        CALL    TDHE
        IF (PSW,IS,ZERO)
            RET
        ENDIF
;
;NOW OFFSET POINTS TO EXP DELTA--SKIP IT !!
;
        LHLD    OFFSET
        INX     H
        INX     H
        SHLD    OFFSET
        LDLD    DLTIPTR
        CALL    TDHE
        IF (PSW,IS,ZERO)
            RET
        ENDIF
ENDLOOP
;
;
        FLDB    VTSUM
        FLDA    XCNT
        FDIV
        FSTA    VTVALUE ;MEAN VT
;
        FLDB    MCASUM
        FLDA    XCNT
        FDIV
        FSTA    MCAVT ;MEAN MCA/VT
;
;
;MEAN MCA/VT = MEAN MCA / MEAN VT
;

```

-49-

FILE: QDCAL.ASM

```

;
; FLDB MCAVALUE
; FLDA UTVALUE
; FDIU
; IF 1, THEN SET = 1
; FPSH
; FATB
; FFOA 1
; FSUB
; FTST AREG.LT
; IF (PSW,IS,ZERO)
;     FPOP
;     FFOA 1
; ELSE
;     FPOP
; ENDF
; FSTA MCAVT
;
;
; COMPUTE THE MEAN VALUE OF % TIME TO PEAK INSP FLOW
; AND MEAN VALUE OF % TIME TO PEAK EXP FLOW
; LHL D, TMPTR
; LXI D,DLTIMES
; CALL TOHE
; IF (PSW,IS,ZERO) ;NONE
;     LXI H,0
;     SHLD PCNUMBERS
;     RET
; ENDF
;
; FFOA 0
; FSTA PKISUM
; FSTA PKESUM
; FSTA TISUM
; LXI H,0
; SHLD PCNUMBERS
; LXI H,DLTIMES
; SHLD ADDR1
;
; LOOP
; MAKE SURE THAT ADDR1+10< TMPTR
; LHL ADDR1
; LXI D,10
; DAD D
; LDED TMPTR
; ORA A
; DSBC D
; IF (PSW,IS,NCARRY) ;THEN ALL DATA NOT THERE
;     RET
; ENDF
;
; ASSUME THAT ADDR1 PTS TO MIN1
; LHL ADDR1
; MOV E,M

```

FILE: QDCAL.ASM

```

        INX     H
        MOV     D,M
        SDED    TMIN1
;GET PKE, MAX, PKI, MIN2 VALUES
        INX     H
        MOV     E,M
        INX     H
        MOV     D,M
        SDED    PKE      ;FOR LAST BREATH
        INX     H
        MOV     E,M
        INX     H
        MOV     D,M
        SDED    TMAX
        INX     H
        MOV     E,M
        INX     H
        MOV     D,M
        SDED    PKI
        INX     H
        SHLD    ADDR1    ;SO NEXT TIME ADDR1 PTS TO MIN1 OF NEXT
        MOV     E,M
        INX     H
        MOV     D,M
        SDED    TMIN2
        INX     H
        MOV     E,M
        INX     H
        MOV     D,M
        SDED    PKE      ;FOR THIS BREATH
;% T TO PEAK INSP FLOW = (PKI-TMIN1) / (TMAX-TMIN1) * 100%
        LHLD    TMAX
        LDOD    TMIN1
        ORA     A
        DSBC    D
        SHLD    TTEMP
        LHLD    PKI
        LDOD    TMIN1
        ORA     A
        DSBC    D
        SHLD    TT2EMP
        FILA    TT2EMP
        FLOT
        FATB
        FILA    TTEMP
        FLOT
        FDIU
        FATB
        FFOA    100
        FMUL
;% T TO PEAK INSP FLOW
        FLDB    PKISUM
        FAOD

```

-51-

FILE: QDCAL.ASM

```

        FSTA     PKISUM
;X T TO PEAK EXP FLOW:= (PKE-TMAX) / (TMIN2-TMAX) * 100 %
        LHLD     PKE
        LOED     TMAX
        ORA      -
        DSBC     0
        SHLD     TTEMP
        LHLD     TMIN2
        LOED     TMAX
        ORA      -
        DSBC     0
        SHLD     TT2EMP
        FILA     TTEMP
        FLOT
        FATB
        FILA     TT2EMP
        FLOT
        FDIU
        FATB
        FFOA     100
        FMUL
;X T TO PEAK EXP FLOW
        FLOB     PKESUM
        FADD
        FSTA     PKESUM
;
;TI = TMAX - TMIN1 / SFREQ
        LHLD     TMAX
        LOED     TMIN1
        ORA      -
        DSBC     0
        SHLD     TTEMP
        FILA     TTEMP
        FLOT
        FATB
        FILA     SFREQ
        FLOT
        FDIU
;AREG = TI
        FLOB     TISUM
        FADD
        FSTA     TISUM
;
        LHLD     PCNUMBERS
        INX      H
        SHLD     PCNUMBERS
ENDLOOP
;
        FLOB     PKISUM
        FILA     PCNUMBERS
        FLOT

```

FILE: QDCAL.ASM

```

      FDIV
      FSTA      PKIAVG
;
      FLD8      PKESUM
      FILA      PCNUMBERS
      FLOT
      FDIV
      FSTA      PKEAVG
;
      FLD8      TISUM
      FILA      PCNUMBER3
      FLOT
      FDIV
      FSTA      TICONTROL

```

```

; VTTICONTROL = ONEHUND / TI
; FLD8      ONEHUND
; FLDA      TICONTROL
; FDIV
; FSTA      VTTICONTROL

```

FF0A
FATB

100;90

```

; NOTE: THIS COMPUTATION MUST BE DONE AFTER MCA COMPUTATION
; BECAUSE HYPOPV AND MCAVALUE ARE AT THE SAME ADDR
; TO CONSERV RAM !!!
;

```

```

; WILL WANT TO PRINT OUT VALUE OF HYPOPNEA (VTLIMIT) = ML

```

```

      LDA      QUALFLG
      IF (A,IS,ZERO) ;QUANT 50 IN ML
          FLDA  VTLIMIT
          FSTA  HYPOPV ;HYPOP IN ML
      ELSE
          ;QUAL 50 PRINT IN TERMS OF %
          FLD8  UTMX
          FFOA  10
          FDIV

```

```

; AREG = CONTROL VALUE IN ML
      FLD8  VTLIMIT ;HYPOP VOLUME IN ML
      FDIV
      FATB
      FFOA  100
      FMUL
      FSTA  HYPOPV ;HYPOP IN % OF CONTROL
      ENDIF
;
;

```

```

      RET
;

```

```

; STARS: .ASCIS

```

```

*****

```

-53-

FILE: QDCAL.ASM

BSBAD: .ASCIS

** BAD ISOVOLUME SCALING CALIBRATION:
TRIED TO SCALE TO 303 ML

RETRY WITH Y-SCALING AT LOWER VALUE

BSBPT: .WORD ONEHUND

BADC: .ASCIS

*** BAD QUALITATIVE CALIBRATION: ***

RC STD = 303

AB STD = 303

DELTAS = 301

MEAN RC = 303

MEAN AB = 303

BADPT:

.WORD RCSTD,ABSTD,ACNT,FRC,FAB

BS1: .ASCIS

BASELINE:

QUALITATIVE ISOVOLUME CALIBRATION:

From 301 deltas:

RC CAL = 303

AB CAL = 303

Breathing Freq = 303 Bths/min

BS3: .ASCIS

BASELINE WITH QUANTITATIVE ISOVOLUME CALIBRATION:

Breathing Freq = 303 Bths/min

BS3PT:

.WORD FREQCONTRL

BS7: .ASCIS

BASELINE WITH GAINS TYPED IN QUANTITATIVELY

Breathing Freq = 303 Bths/min

-54-

FILE: QDCAL.ASM

BS7PT:

.WORD FREQCONTRL

V1: .ASCIS

QUALITATIVE CALIBRATION:

From 301 deltas:

RC CAL = 303

AB CAL = 303

Breathing Freq = 303 Bths/min

V7: .ASCIS

BASELINE WITH GAINS TYPED IN QUALITATIVELY:

From 301 deltas:

RC CAL = 303

AB CAL = 303

Breathing Freq = 303 Bths/min

V2: .ASCIS

From 301 breaths:

Mean MCA/VT = 303

Mean XT PK INSP FL = 303 %

Mean XT PK E = FL = 303 %

Mean TI = 303 sec

HYP: .ASCIS

Hypopnea volume = 303

AHYP: .ASCIS ml

RHYP: .ASCIS % of baseline

HYPPT: .WORD HYPOPV

BS1:=""

V7PT:

V1PT: .WORD XCNT,RCCAL,ABCAL

.WORD FREQCONTRL

V2PT: .WORD

XCNT

.WORD MCAVT

FILE: QDCAL.ASM

```

        .WORD    PKIAVG,PKEAVG
        .WORD    TICONROL

```

RESCALE:

```

        FLDB      UTMX
        FFDA      10
        FDIY
        FSTA      ONEHUND ;THIS THE THE VOLUME SCALING TO
;COMPUTE THE FREQ SO CAN COMPUTE THE UMIN FOR CONTROL
        FILA      SECONDS
        FLOT
        FATB
        FFCA      60
        FDIY
        FATB
        FILA      ELPMIN
        FLOT
        FADD      ;AREQ = # MINUTES FOR COLLECTION
        FPSH

```

```

        LHLD      MAXCNT ;# OF DELTAS
        SRAR      H
        RARR      L ;MAXCNT/2 == # BREATHS
        SHLD      ITEMP
        FILA      ITEMP
        FLOT
        FATB      ;# BREATHS
        FPOP      ;# MINUTES
        FDIY
        FSTA      FREQCONTRL

```

```

        FLDB      ONEHUND ;100 % VOLUME
        FLDA      FREQCONTRL ;FREQ
        FMUL
        FSTA      UMINCONTRL ;UMIN OF CONTROL

```

SMMEAN = MEAN VT IN CUS

```

        FLDB      SMMEAN
        FLDA      CCS
        FMUL
        FSTA      MEAN ;MEAN VT IN ML

```

```

        LDA      QUALFLG
        IF (.A,IS,NZERO) ;QUALITATIVE

```

-56-

FILE: QDCAL.ASM

```

; SO INIT VTLIMIT TO ONEHUND * VTPCL%
    FLDB VTPCL
    FFOA 100
    FDIV
    FLDB ONEHUND
    FMUL
    FSTA VTLIMIT ;FOR HYPONNEA DETECTION
;
ELSE ;QUANTITATIVE, USE MEAN VT INSTEAD OF ONEHUND
    FLDB VTPCL
    FFOA 100
    FDIV
    FLDB MEAN ;MEAN VT IN ML
    FMUL
    FSTA VTLIMIT ;FOR HYPONNEA DETECTION
ENDIF
;
    FLDB UTMX
    FFOA 10
    FDIV
    FLDB MEAN
    FSWP
    FDIV ;RATIO OF (UTMX/10) / MEAN VT
    FSTA MEAN
;RCCAL = CURRENT RC GAIN (RCGNF) * MEAN
;ABCAL = CURRENT AB GAIN (ABGNF) * MEAN
    FLDA MEAN
    FLDB RCGNF ;CAL FROM INITIAL CALCS
    FMUL
    FSTA RCCAL
;
    FLDA MEAN
    FLDB ABGNF ;CAL FROM INITIAL CALCS
    FMUL
    FSTA ABCAL
;
;NOW THE CAL FACTORS ARE SCALED SO UTMX/10 WILL APPEAR AS 100 % DURING
; LONG TERM
    RET

.EXTERN QD2
.LOC QD2
RCMCA: .BLKB 2
ABMCA: .BLKB 2
SMMCA: .BLKB 2
ADDR1: .BLKB 2
;
MCAVALUE:
HYPOPV:
    .BLKB 4
;
;
;

```

-57-

FILE: QDCAL.ASM

```
UTVALUE: .BLKB 4
MCAVT: .BLKB 4
UTSUM: .BLKB 4
MCASUM: .BLKB 4
PKIavg: .BLKB 4
PKEavg: .BLKB 4
TISUM: .BLKB 4
      .RELOC
;
      .END
```

FILE: VSETUP.ASM

58

```

; IDENT VSETUP
; INSERT SP80.ASM
; INSERT FPMAC.SRC
; INSERT CALCOM.ASM
; INSERT CKCOM
; INSERT SELCOM
; INSERT BAGCOM
; INSERT VARCOM.CCC
; INSERT SYCOM
; INSERT HRDCOM
;
; .EXTERN APBP,ZALL,AUDEN
;
; .EXTERN RTON,RTOF,RTICK
; .EXTERN TDHE
; .EXTERN DERIV5
; .EXTERN PTRUPD,GETP2,FIND,ROUND
;
; .EXTERN SPON,SPOF
;
; .ENTRY SETUP
;
; CR = 00H
; FINIT
;
; SETUP:
;
; ;SETUP THE CLOCK FOR VALIDATION
;
; ;SAMPLE AT 20 PTS PER SEC
; PUSH H
; ; INIT REAL-TIME BUFFER POINTERS
; ; MODIFIED 09-24-81
; MVI A,3
; STA WAITFLG
;
; XRA A
; STA WRAPF
; STA SECFLG
; CMA
; STA BFLG
; XRA A
; STA VFRSTMX
; LXI H,0
; SHLD ELPTM
; CHLD ELPMIN
; SHLD SECONDS
; SHLD CPPTR

```

-59-

FILE: VSETUP.ASM

```

    LXI    H,0
    SHLD   RIPT
    SHLD   ODRIPT
    SHLD   SPTMP
    SHLD   PNSUM
;
;
    LXI    H,8      ;SKIP THE 1ST 4 DATA POINTS
    SHLD   ROPT
;
    LXI    H,TMBUFF
    MVI    M,0
    INX    H
    MVI    M,0
    LXI    H,0
    SHLD   TIME
; INIT VARIABLES FOR PICKING MAX AND MINS
    XRA    A
    STA    PTF      ;START BY LOOKING FOR MAX PT
    STA    SLOPEF   ;INIT MAX NOT FOUND
;
    XRA    A
    STA    INTFLG
    CMA
    STA    GFLG     ;INDICATES CLOCK ON
    CALL   AUDEN
    CALL   APBP
    CALL   ZALL
;
;
    LXI    H,IRSRV
    MVI    A,20     ;CLOCK RATE
    CALL   RTON     ;TURN ON CLOCK
;
    LDA    BAG
    IF (.A,IS,ZERO) ;USING SPIROMETER
        CALL SPON
    ENDIF
;
;
    POP    H
    RET
;
;
CLKSTP:
    ;STOP DATA COLLECTION
    ;TURN CLOCK OFF
;
;MODIFIED 09-24-81
;
    CALL   RTOF     ;TURN OFF CLOCK
    LDA    BAG

```


-61-

FILE: VSETUP.ASM

```

        MVI    A,0FFH
        STA    SLOPEF
    ENDIF
;
    CALL    DERIV5
;
    LDA    VFRSTMX
    IF (.A,IS,ZERO) ;HAVE NOT FOUND 1ST MAX
        LHL    ROPT
        CALL    FIND
        IF (PSW,IS,NCARRY) ;NOT MAX
            LHL    ROPT
            LXI    B,RSIZE
            CALL    PTRUPD
            SHLD    ROPT
            JMP    EXIT
        ENDIF
        ;MAX FOUND
        LDA    VFRSTMX
        CMA
        STA    VFRSTMX
        LDA    PTF
        CMA
        STA    PTF
        SHLD    SPTEMP ;TEMP CRIT PT
        SDED    RCTEMP ;CORRESPONDING RC PT
        SBCD    ABTEMP ;CORRESPONDING AB PT
;
        LHL    ROPT
        LXI    B,RSIZE
        CALL    PTRUPD
        SHLD    ROPT
        JMP    EXIT
    ENDIF
;
;
        ;INIT MAX FOUND
        LHL    ROPT
        CALL    FIND
;
;UPDATE ROPT FOR NEXT TIME
;
        PUSH    PSW
        PUSH    D
        PUSH    B
        PUSH    H
;
        LHL    ROPT
        LXI    B,RSIZE
        CALL    PTRUPD
        SHLD    ROPT
        POP     H
        POP     B

```

FILE: USETUP.ASM

```

        POP    D
        POP    PSW

;
;
;FIND RETURNED C SET IF CP FOUND
;AND HL HOLDS SP PT VALUE
; AND DE HOLDS RC PT VALUE
; AND BC HOLDS AB PT VALUE
;
        IF (PSW,IS,CARRY)
        SHLD   SPVALUE
        SDED   RCVALUE
        SBCD   ABVALUE
;
;
        LDA    PTF
        IF (.A,IS,ZERO) ;THEN MAX FOUND
            XRA    A
            CMA
            STA    BFLG
            LXI    D,SPMAX
            LHLD   CPPTR
            DAD    D
;
            LDA    BAG
            IF (.A,IS,ZERO) ;THEN USING SPIROMETER, AND
                                ;SPVALUE HOLDS SPIROMETER VALUE
            LDED   SPVALUE
            ELSE    ;USING SPIROBAG,SO FIXCU HOLDS VALUE
            LDED   FIXCU
            ENDIF
;
            MOV    M,E
            INX    H
            MOV    M,D
;
            LHLD   CPPTR
            LXI    D,RCMAX
            DAD    D
            LDED   RCVALUE
            MOV    M,E
            INX    H
            MOV    M,D
;
            LHLD   CPPTR
            LXI    D,ABMAX
            DAD    D
;
;
;IF VALIDATING AND IF USING SPIROBAG
; THEN SPVALUE HOLDS THE SUM MAX
        LDA    BAG
        IF (.A,IS,NZERO) ;THEN USING SPIROBAG

```

```

;
L1001: LD ED SPVALUE ;SUM VALUE
        JMP L1001
        ENDIF

        LD ED ABVALUE

MOV M,E
INX H
MOV M,D
LHLD MAXCNT
INX H
SHLD MAXCNT
LHLD CPPTR
INX H
INX H
SHLD CPPTR
LHLD MAXCNT
LXI D,BRTHMX
ORA A
DSBC D
IF (PSW,IS,ZERO) ;THEN BUFFER FULL
    JMP CLKSTP
ENDIF

ELSE
    LHLD CPPTR
    LXI D,SPMIN
    DAD D

    LDA BAG
    IF (.A,IS,ZERO) ;THEN USING SPIROMETER, AND
                    ;SPVALUE HOLDS SPIROMETER VALUE
    LD ED SPVALUE
    ELSE           ;USING SPIROBAG,SO MIN = 0
    LXI D,0
    ENDF

MOV M,E
INX H
MOV M,D

LHLD CPPTR
LXI D,RDMIN
DAD D
LD ED RCVALUE
MOV M,E
INX H

MOV M,D

LHLD CPPTR
LXI D,ABMIN
DAD D

```

FILE: VSETUP.ASM

```

LHLD DATA
MOV A,H
CMA
ANI 0FH
MOV H,A
MOV A,L
CMA
MOV L,A
SHLD DATA
;
ENDIF
RET
;
;
;
;
;
;
;
PRNG:
;
;HL HOLDS THE RING INPUT PTR
;DE HOLDS ADDR OF RING
;
DAD D
LDED DATA
MOV M,E
INX H
MOV M,D
RET
;
.END

```

FILE: QDCSTATS.ASM

```

        .IDENT QDSTATS
        .INSERT FPMAC.SRC
        .INSERT SP80
        FINIT
        .INSERT VTCCOM
        .INSERT CALCOM
        .INSERT OLQCOM
        .INSERT CKCOM
        .INSERT VARCOM.CCC
        .INSERT SYCOM
        .INSERT QD1.ASM
;
;
        .EXTERN TDHE,ROUND
;
;
        .ENTRY PRTQDC
        .ENTRY CMPSTATS
;
PRTQDC:
        ;THIS IS THE ENTRY POINT FROM LONG TERM
        CALL    CMPSTATS
        RC
        FLDB    RCSTD
        FLDA    ABSTD
        FDIW
        FSTA    ABCAL
        FPRN    QDCV,QDCPT
        RET
;
QDCV: .ASCIS
QDC = QD3:  # DELTAS = QD1
          MEAN  STD    COV
RC:  QD3      QD3      QD3
AB:  QD3      QD3      QD3
;
QDCPT: .WORD   ABCAL,XCNT
        .WORD   FRC,RCSTD,COVRC
        .WORD   FAB,ABSTD,COVAB
;
;
NODELTS: .ASCIS
NO QDC DELTAS AVAILABLE
;
;
;
;
CMPSTATS:
;RCDATA AND ABDATA HOLD THE DELTAS FOR RC AND AB
;
;COMPUTE THE MEAN AND STD OF RC, AB AND RC+AB
;

```

-66-

FILE: QDCSTATS.ASM

```

        LHL DLTIPTR
        SHLD DLTOFF ;SAVE IT INCASE IT CHANGES IN THE
                    ;MIDDLE OF THIS DURING LONG TERM
;
;
        LDA DLTFULL
        IF (.A,IS,ZERO) ;RING NOT FULL YET
            LXI H,0
            SHLD OFFSET
            LD ED,DLTOFF
            CALL TDHE
            IF (PSW,IS,ZERO) ;THEN NO DELTAS
                LXI H,NODELTS
                CALL TXTYP
                STC
                RET
            ENDIF
        ELSE ;RING IS FULL
            LHL DLTOFF ;POINTS TO NEXT PLACE TO PUT, SO WILL BE
                    ;THE OLDEST VALUE
            SHLD OFFSET
        ENDIF
;
;
;1ST TIME THRU THE DATA COMPUTE MEAN AND STD OF JUST SUM
;
;2ND TIME THUR THE DATA ONLY USE DATA WHERE SUM IS WITHIN +-QDCSTD OF MEAN
; OF SUM
;
        FFOA 0
        FSTA SMSUM
        FSTA SMXXSUM
        FSTA XCNT
;
; LOOP
        LHL D,OFFSET
        LXI D,RCDATA
        DAD D
        MOV E,M
        INX H
        MOV D,M
        SDED IRCV
;
        LHL D,OFFSET
        LXI D,ABDATA
        DAD D
        MOV E,M
        INX H
        MOV D,M
        SDED IABV
;

```

-67-

FILE: QDCSTATS.ASM

```

        CALL    SUMMERS
;
        LHLD    OFFSET
        INX     H
        INX     H
        SHLD    OFFSET
        LXI     D,1000
        CALL    TDHE
        IF (PSW,IS,ZERO)
            LXI     H,0
            SHLD    OFFSET
        ENDIF
;
;
;QUIT WHEN OFFSET = DLTOFF
        LHLD    OFFSET
        LDED    DLTOFF
        CALL    TDHE
        IF (PSW,IS,ZERO)
            RET     ;EXIT LOOP
        ENDIF
ENDLOOP
;COMPUTE MEAN AND STD OF SUM
        FLDB    SMSUM
        FLDA    XCNT
        FDIV
        FSTA    SMMEAN
;
        FLDB    SMSUM
        LDA     SMSUM
        FMUL
        FATB
        FLDA    XCNT
        FDIV
        FLDB    SMXXSUM
        FSUB
        FATB
        FLDA    XCNT
        FDIV
        FSQR
        FSTA    SMSTD
;HIGH LIMIT = SMMEAN + QDCSTD*SMSTD
;LOW LIMIT = SMMEAN - QDCSTD*SMSTD
;
        FLDB    QDCSTD
        FLDA    SMSTD
        FMUL
        FLDB    SMMEAN
        FADD
        FSTA    SMHIGH
;
        FLDB    QDCSTD

```

FILE: QDCSTATS.ASM

```

FLDA    SMSTD
FMUL
FLDB    SMMEAN
FSUB
FSTA    SMLOW

```

```

;
;NOW COMPUTE MEAN AND STD OF RC AND AB FOR ALL PAIRS WHERE
; RC+AB < SMHIGH AND RC+AB > SMLOW
;
;

```

```

LDA      DLTFULL
IF (.A,IS,ZERO) ;RING NOT FULL YET
    LXI    H,0
    SHLD   OFFSET
ELSE
    LHL    DLTFF    ;RING IS FULL
                  ;POINTS TO NEXT PLACE TO PUT, SO WILL BE
                  ;THE OLDEST VALUE
    SHLD   OFFSET
ENDIF

```

```

;
;
FFOA     0
FSTA     RCSUM
FSTA     RCXXSUM
FSTA     ABSUM
FSTA     ABXXSUM
FSTA     XCNT

```

```

;
LOOP

```

```

    LHL    OFFSET
    LXI    D,RCDATA
    DAD    D
    MOV    E,M
    INX    H
    MOV    D,M
    SDED   IRCV

```

```

    LHL    OFFSET
    LXI    D,ABDATA
    DAD    D
    MOV    E,M
    INX    H
    MOV    D,M
    SDED   IABV

```

```

;
    CALL   RASUMMERS
;

```

```

    LHL    OFFSET
    INX    H
    INX    H
    SHLD   OFFSET
    LXI    D,1000

```

-69-

FILE: QDCSTATS.ASM

```

        CALL    TDHE
        IF (PSW,IS,ZERO)
            LXI    H,0
            SHLD   OFFSET
        ENDIF
;
;
;QUIT WHEN OFFSET = DLTOFF
        LHL    OFFSET
        LOED   DLTOFF
        CALL   TDHE
        IF (PSW,IS,ZERO)
            RET    ;EXIT LOOP
        ENDIF
ENDLOOP
;
;IF XCNT < 2, THEN NO GOOD DATA
        FLDB    XCNT
        FFOA    2
        FSUB
        FTST    AREG,LT
        IF (PSW,IS,ZERO)
            LXI    H,NODELTS
            CALL   TXTYP
            STC
            RET
        ENDIF
;COMPUTE MEAN OF RC AND AB
        FLDB    RC SUM
        FLDA    XCNT
        FDI V
        FSTA    FRC
;
        FLDB    ABSUM
        FLDA    XCNT
        FDI V
        FSTA    FAB
;
;COMPUTE STD OF RC AND STD OF AB
;
        FLDB    RC SUM
        FLDA    RC SUM
        FMUL
        FATB
        FLDA    XCNT
        FDI V
        FLDB    RCXXSUM
        FSUB
        FATB
        FLDA    XCNT
        FDI V
        FSQR

```

-70-

FILE: QDCSTATS.ASM

```

;
    FSTA    RCSTD
;
    FLDB    ABSUM
    FLDA    ABSUM
    FMUL
    FATB
    FLDA    XCNT
    FDIV
    FLDB    ABXXSUM
    FSUB
    FATB
    FLDA    XCNT
    FDIV
    FSQR
    FSTA    ABSTD
;MAKE VALUES ML
    FLDB    FRC
    FLDA    CCS
    FMUL
    FSTA    FRC
;
    FLDB    FAB
    FLDA    CCS
    FMUL
    FSTA    FAB
;
    FLDB    ABSTD
    FLDA    CCS
    FMUL
    FSTA    ABSTD
;
    FLDB    RCSTD
    FLDA    CCS
    FMUL
    FSTA    RCSTD
;
;COV = STD/MEAN
    FLDB    RCSTD
    FLDA    FRC
    FDIV
    FATB
    FFOA    100
    FMUL
    FSTA    COVRC
;
    FLDB    ABSTD
    FLDA    FAB
    FDIV
    FATB
    FFOA    100
    FMUL

```

-71-

FILE: QDCSTATS.ASM

```

        FSTA    COVAB
        ORA     A
        RET

;
;
;
RASUMMERS:
        FILA    IRCV
        FLOT
        FSTA    FRC

        FILA    IABV
        FLOT
        FSTA    FAB

;
;DO NOT USE IF FRC+FAB > SMHIGH OR IF FRC+FAB < SMLOW
        FLDB    FRC
        FLDA    FAB
        FADD
        FATB
        FLDA    SMHIGH
        FSUB
        FTST    AREG,GT
        IF (PSW,IS,ZERO)
            RET
        ENDIF

;
        FLDB    FRC
        FLDA    FAB
        FADD
        FATB
        FLDA    SMLOW
        FSUB
        FTST    AREG,LT
        IF (PSW,IS,ZERO)
            RET
        ENDIF

;
;
;
        FLDA    FRC
        FLDB    RCSUM
        FADD
        FSTA    RCSUM

;
        FLDB    FRC
        FLDA    FRC
        FMUL
        FLDB    RCXXSUM
        FADD
        FSTA    RCXXSUM

;
        FLDA    FAB

```

-72-

FILE: QDCSTATS.ASM

```
FLDB  ABSUM
FADD
FSTA  ABSUM
;
FLDB  FAB
FLDA  FAB
FMUL
FLDB  ABXXSUM
FADD
FSTA  ABXXSUM
```

```
;
;
;
FLDB  XCNT
FFOA  1
FADD
FSTA  XCNT
;
RET
```

SUMMERS:

```
FILA  IRCV
FLOT
FATB
FILA  IABV
FLOT
FADD
FSTA  FSUM

FLDA  FSUM
FLDB  SMSUM
FADD
FSTA  SMSUM
;
FLDA  FSUM
FLDB  FSUM
FMUL
FLDB  SMXXSUM
FADD
FSTA  SMXXSUM
;
;
FLDB  XCNT
FFOA  1
FADD
FSTA  XCNT
RET
;
```

-73-

FILE: QDCSTATS.ASM

;

.END

FILE: CFIND.ASM

```

;
; .IDENT CFIND
; .INSERT SP80.ASM
; .INSERT CALCOM
; .INSERT SELCOM
; .INSERT EFFCOM
; .INSERT DRVCOM
; .INSERT SIOCOM
; .INSERT FPMAC.SRC
; .INSERT PKCOM.ASM
; .ENTRY FIND,GETP2,DERIV5
; .EXTERN TDHE,PTRUPD
; .INIT

FIND:
;
; HL = OFFSET OF CURRENT PT ON ENTRANCE
;
; ON EXIT: IF CP FOUND, THEN CARRY SET AND
;           HL = SPIROMETER PT
;           DE = RIB CAGE PT
;           BC = ABDOMEN PT
;           SHLD CUROFF
;
;           LHLD CUROFF
;           LXI D,SPBUFF
;           DAD D
;           MOV E,M
;           INX H
;           MOV D,M
;           SDED CURSP
;
;           LHLD CUROFF
;           LXI D,RCBUFF
;           DAD D
;           MOV E,M
;           INX H
;           MOV D,M
;           SDED CURRC
;
;           LHLD CUROFF
;           LXI D,ABBUFF
;           DAD D
;           MOV E,M
;           INX H
;           MOV D,M
;           SDED CURAB
;
;           LHLD CUROFF
;           LXI D,TMBUFF
;           DAD D
;           MOV E,M
;           INX H
;           MOV D,M

```

FILE: CFIND.ASM

```

        SDED    CURTM
;
;
        CALL    GTSIGN    ;CHECK SIGNS OF DERIVATIVES
;
;
        LDA     PTF
        IF (.A.IS.NZERO)    ;LOOK FOR MIN
            LHL    SPTMP
            LDED    CURSP
            ORA     A
            DSBC    0
            IF (PSW.IS.NCARRY) ;NEW TMP MIN FOUND
                LHL    CURSP
                SHLD    SPTMP
                LHL    CURRC
                SHLD    RCTEMP
                LHL    CURAB
                SHLD    ABTEMP
                LHL    CUROFF
                SHLD    OFFTEMP
                LHL    CURTM
                SHLD    TMTEMP
;1. ADD CURRENT RCDERV PT TO RC COUNTS
;2. ADD RCTENT TO RC COUNTS
;3. RCTCNT <== 0
;4. ADD CURRENT ABDERV TO AB COUNTS
;5. ADD ABTCNT TO AB COUNTS
;6. ABTCNT <== 0
            CALL    UPDT1

            ORA     A
            RET
        ELSE
            ;POSSIBLE CP SO CHECK SLOPE BETWEEN
            ;CURRNT PT AND TEMP PT
            LHL    CURSP
            LDED    SPTMP
            ORA     A
            DSBC    0
            ;NOW HL HOLDS DIFFERENCE
            LDED    MINCC
            ORA     A
            DSBC    0
            IF (PSW.IS.NCARRY) ;THEN SPTMP IS CP
                LHL    OFFTEMP
                SHLD    CPOFF
                LHL    SPTMP
                PUSH    H
                LHL    RCTEMP
                PUSH    H
                LHL    ABTEMP
                PUSH    H

```

FILE: CFIND.ASM

```

        LHL D CURSP
        SHL D SPTMP
        LHL D CURRC
        SHL D RCTMP
        LHL D CURAB
        SHL D ABTMP
        LHL D CUROFF
        SHL D OFFTMP
        LHL D TMTEMP
        SHL D CPTIME ;TIME OF THIS CP
;
        LHL D CURTM
        SHL D TMTEMP ;NEW TEMP
;
;1. STORE RC COUNTS
;2. RC COUNTS <= RCTCNT
;3. RCTCNT <= 0
;4. STORE AB COUNTS
;5. AB COUNTS <= ABTCNT
;6. ABTCNT <= 0
;
        CALL UPDT2
;
        POP B ;AB VALUE
        POP D ;RC VALUE
        POP H ;SP VALUE
        STC ;SET CARRY
        RET
    ELSE
        ;NOT A LARGE ENOUGH SLOPE
;
;1. ADD RCDERV TO RCTCNT
;2. ADD ABDERV TO ABTCNT
        CALL UPDT3
        ORA A
        RET
    ENDIF
ENDIF
ELSE ;LOOK FOR MAX
    LHL D SPTMP
    LD D CURSP
    XCHG
    ORA A
    DSBC D
    IF (PSW,IS,NCARRY) ;;THEN CURRENT >= SPTMP
        LHL D CURSP
        SHL D SPTMP
        LHL D CURRC
        SHL D RCTMP
        LHL D CURAB
        SHL D ABTMP
        LHL D CUROFF
        SHL D OFFTMP

```

FILE: CFIND.ASM

```

        LHL    CURTM
        SHL    TMTEMP

        CALL    UPDT1
        ORA    A
        RET

    ELSE ;POSSIBLE CP
        LHL    SPTMP
        LDE    CURSP
        ORA    A
        DSB    0
;NOW HL = DIFF
        LDE    MINC
        ORA    A
        DSB    0
        IF (PSW,IS,NCARRY) THEN SPTMP IS CP
            LHL    OFFTEMP
            SHL    CPOFF
            LHL    SPTMP
            PUSH    4
            LHL    RCTMP
            PUSH    4
            LHL    ABTEMP
            PUSH    4

            LHL    TMTEMP
            SHL    OPTIME ;TIME OF CP

            LHL    CURSP
            SHL    SPTMP
            LHL    CURRC
            SHL    RCTMP
            LHL    CURAB
            SHL    ABTEMP
            LHL    CUROFF
            SHL    OFFTEMP
            LHL    CURTM
            SHL    TMTEMP

            CALL    UPDT2
            POP     B        ;AB VALUE
            POP     D        ;RC VALUE
            POP     H        ;SP VALUE
            STC
            RET
        ELSE ;NOT LARGE ENOUGH SLOPE
            CALL    UPDT3
            ORA    A
            RET
    ENDIF
ENDIF
ENDIF

```

FILE: CFIND.ASM

```

;
;
;
;GETP2:
;SEE IF AT LEAST 5 POINTS IN RING BUFFER
;
;    LHL0    RIPT
;    LXI     D,10
;    ORA     A
;    DSBC    D
;    IF (PSW.IS.CARRY) THEN < 5 PTS
;        STC
;    ELSE
;        ORA     A
;    ENDIF
;    RET
;
;

```

UPOT1:

```

;1. ADD CURRENT ABS(RCDERV) PT TO RC COUNTS
;2. ADD RCTCNT TO RC COUNTS
;3. RCTCNT <= 0
;4. ADD CURRENT ABS(ABOERV) TO AB COUNTS
;5. ADD ABTCNT TO AB COUNTS
;6. ABTCNT <= 0
;
;
;

```

;*****

;NOTE : DO NOT SEPEATAE INPHASE AND OUT "OF PHASE FOR AB"

;*****

```

;
;    LHL0    ABSRCD
;    LOED    RCIN
;    DAD     D
;    SHLD    RCIN
;
;

```

```

;
;    LHL0    RCIN
;    LOED    RCTCNT
;    DAD     D
;    SHLD    RCIN
;
;

```

```

;
;    LXI     H,0
;    SHLD    RCTCNT
;
;

```

```

;
;    LHL0    ABSABD
;    LOED    ABIN
;    DAD     D
;
;

```

FILE: CFIND.ASM

-79-

```

        SHLD    ABIN
;
;
        LHL    ABIN
        LD     ABTICNT
        DAD    D
        SHLD    ABIN
;
        LXI    H,0
        SHLD    ABTICNT
;
;SUM: DO NOT SEPERATE INPHASE AND OUT OF PHASE VALUES
;      ADD THEM ALL
;
        LHL    ABSSMD
        LD     SMIO
        DAD    D
        SHLD    SMIO
;
;
        LHL    SMIO
        LD     SMTCNT
        DAD    D
        SHLD    SMTCNT
;
;
        LXI    H,0
        SHLD    SMTCNT
;
;FOR PEAK FLOW STUFF:
;SMDERV = CURRENT FLOW, ABSSMD = ABS(SMDERV)
;
;IF ON INSP SIDE:, LOOKING FOR MAX (PTF=0)
        LDA     PTF
        IF (.A.IS,ZERO) ;ON INSP SIDE
;IF CURRENT FLOW > 0, OR IF ABSSMD = SMDERV
        LHL    ABSSMD
        LD     SMDERV
        CALL    TOHE
        IF (PSW,IS,ZERO) ;THEN CURRENT FLOW > 0
;IF CURRENT FLOW > TEMPK
        LHL    TEMPK
        LD     SMDERV
        ORA     A
        OSBC    D
        IF (PSW,IS,CARRY) ;CURRENT FLOW > TEMPK
                LHL    SMDERV
                SHLD    TEMPK
                LHL    CURTM
                SHLD    PKTIME
        ENDIF
        ENDIF
;IF TTPOS > TEMPK

```

FILE: CFIND.ASM

```

        LHL D      TEMPK
        LOD D      TTPOS
        OR A
        DSB C      D
        IF (PSW, IS, CARRY) ; TTPOS > TEMPK
            LHL D      TTPOS
            SHL D      TEMPK
            LHL D      TTPTM
            SHL D      PKTIME
        ENDIF
;
        LXI H, 0
        SHL D      TTPOS
        SHL D      TTNEG
        LHL D      CURTM
        SHL D      TTPTM
        SHL D      TTNTM
;
        ELSE          ; ON THE EXP SIDE (LOOKING FOR MIN
;
; IF CURRENT FLOW < 0, OR IF ABS(SMD) NOT= SMDERV
        LHL D      ABS(SMD)
        LOD D      SMDERV
        CALL      TDHE
        IF (PSW, IS, NZERO) ; THEN CURRENT FLOW = 0
; IF ABS(CURRENT FLOW) > TEMPK
            LHL D      TEMPK
            LOD D      ABS(SMD)
            OR A
            DSB C      D
            IF (PSW, IS, CARRY) ; ABS(CURRENT FLOW) > TEMPK
                LHL D      ABS(SMD)
                SHL D      TEMPK
                LHL D      CURTM
                SHL D      PKTIME
            ENDIF
        ENDIF
; IF TTNEG > TEMPK
        LHL D      TEMPK
        LOD D      TTNEG
        OR A
        DSB C      D
        IF (PSW, IS, CARRY) ; TTNEG > TEMPK
            LHL D      TTNEG
            SHL D      TEMPK
            LHL D      TTNTM
            SHL D      PKTIME
        ENDIF
;
        LXI H, 0
        SHL D      TTPOS
        SHL D      TTNEG
        LHL D      CURTM

```

FILE: CFIND.ASM

```

                SHLD    TTPTM
                SHLD    TTNTM
;
;
                ENDIF

                RET

;
;
UPDT2:
;
;1. STORE RC COUNTS
;2. RC COUNTS <== RCTCNT
;3. RCTCNT <== 0
;4. STORE AB COUNTS
;5. AB COUNTS <== ABTCNT
;6. ABTCNT <== 0
        LHLD    RCIN
        SHLD    RCINVOL
;
;
        LHLD    RCTCNT
        SHLD    RCIN
;
;
        LXI     H,0
        SHLD    RCTCNT
;
        LHLD    ABIN
        SHLD    ABINVOL
;
;
        LHLD    ABTCNT
        SHLD    ABIN
;
;
        LXI     H,0
        SHLD    ABTCNT
;
;SUM: DO NOT SEPERATE INPHASE AND OUT OF PHASE
        LHLD    SMIO
        SHLD    SMVOL
;
;
        LHLD    SMTCNT
        SHLD    SMIO
;
;
        LXI     H,0
        SHLD    SMTCNT
;
;FOR PEAK FLOW STUFF:
;SMDERV = CURRENT FLOW - SMD = ABS(SMDERV)

```

FILE: CFIND.ASM

```

;
; IF ON INSP SIDE, LOOKING FOR MAX (PTF=0)
    LDA     PTF
    IF (A.IS.ZERO) ; ON INSP SIDE
; TEMPK IS THE PEAK FLOW, AND PKTIME IS THE TIME OF IT
; STORE THE TIME OF THE PEAK FLOW
        CALL PKSTORE
        LHL TTNEG ; NOW WILL BE ON EXP SIDE
        SHLD TEMPK
        LHL TTNTM
        SHLD PKTIME
        LXI 4,0
        SHLD TTPOS
        SHLD TTNEG
        LHL CURTM
        SHLD TTPTM
        SHLD TTNTM
    ELSE ; ON EXP SIDE
; TEMPK IS THE PEAK FLOW, AND PKTIME IS THE TIME OF IT
; STORE THE TIME OF THE PEAK FLOW
        CALL PKSTORE
        LHL TTPOS ; NOW WILL BE ON INSP SIDE
        SHLD TEMPK
        LHL TTPTM
        SHLD PKTIME
        LXI 4,0
        SHLD TTPOS
        SHLD TTNEG
        LHL CURTM
        SHLD TTPTM
        SHLD TTNTM
    ENDIF

;
    RET
;
;
;
UPDT3:
;
; 1. ADD ABS(RCDERV) TO RCTCNT
; 2. ADD ABS(ABDERV) TO ABTCNT
;
        LHL ABSRCD
        LDED RCTCNT
        DAD D
        SHLD RCTCNT
;
        LHL ABSABD
        LDED ABTCNT
        DAD D
        SHLD ABTCNT

```

FILE: CFIND.ASM

;SUM: DO NOT SEPERATE INPHASE AND OUT OF PHASE

```

      LHL D  ABSSMD
      LDE D  SMTCNT
      DAD  D
      SHL D  SMTCNT

```

;FOR PEAK FLOW STUFF:

;SMDERV = CURRENT FLOW. ABSSMD = ABS(SMDERV)

;IF CURRENT FLOW > 0

```

      LHL D  SMDERV
      LDE D  ABSSMD
      CALL TDHE

```

IF (PSW.IS,ZERO)

;THEN CURRENT FLOW > 0

;IF CURRENT FLOW > TTPOS

```

      LHL D  TTPOS
      LDE D  SMDERV
      ORA  A
      OSBC D

```

IF (PSW.IS,CARRY)

```

      LHL D  SMDERV
      SHL D  TTPOS
      LHL D  CURTM
      SHL D  TTPTM

```

ENDIF

ELSE

;IF ABS(CURRENT FLOW) > TTNEG

```

      LHL D  TTNEG
      LDE D  ABSSMD
      ORA  A
      OSBC D

```

IF (PSW.IS,CARRY)

```

      LHL D  ABSSMD
      SHL D  TTNEG
      LHL D  CURTM
      SHL D  TTNTM

```

ENDIF

ENDIF

RET

;PKSTORE:

```

      LHL D  PKTIME
      SHL D  PKVALUE
      RET

```

FILE: CFIND.ASM

```

;
GTSIGN:
    LHLD    RCDERV    ;CURRENT RC DERIV VALUE
    MOV     A,H
    ANI     80H
    STA     RCSIGN
;
    LHLD    ABDERV    ;CURRENT AB DERIV VALUE
    MOV     A,H
    ANI     80H
    STA     ABSIGN
;
    LHLD    SMDERV
    MOV     A,H
    ANI     80H
    STA     SMSIGN
;
;
    MOV     B,A
    LDA     RCSIGN
    IF (.A,EQ,.8)    ;THEN RC INPHASE W/ SUM
        XRA     A
        CMA
        STA     RCINPHS
    ELSE
        XRA     A
        STA     RCINPHS
    ENDIF
;
    LDA     SMSIGN
    MOV     B,A
    LDA     ABSIGN
    IF (.A,EQ,.8)    ;THEN AB INPHASE W/ SUM
        XRA     A
        CMA
        STA     ABINPHS
    ELSE
        XRA     A
        STA     ABINPHS
    ENDIF
    RET

```

DERIV5:

```

;*****
;DERIV USING 9 POINTS

```

FILE: CFIND.ASM

;*****

;DERIVATIVE USING 5 POINTS

;SRCOFST HOLDS OFFSET TO GET Y5 FROM RAW DATA BUFFERS

;ASSUME: RCY1,RCY2,RCY3,RCY4, ABY1,ABY2,ABY3,ABY4 ALREADY SET

```

;
    LXI    D,RCBUFF
    LHLD   SRCOFST
    DAD    D
    MOV    E,M
    INX    H
    MOV    D,M
    SDED   RCY9

```

```

;
    LHLD   SRCOFST
    LXI    D,ABBUFF
    DAD    D
    MOV    E,M
    INX    H
    MOV    D,M
    SDED   ABY9

```

```

;
    LXI    B,RSIZE ;UPDATE FOR NEXT TIME
    LHLD   SRCOFST
    CALL   PTRUPD
    SHLD   SRCOFST

```

;DERIV = (9-1)*4 + (8-2)*3 + (7-3)*2 + (6-4)

```

;
    LHLD   RCY9
    LDED   RCY1
    ORA    A
    DSBC   D
    DAD    H          ;(RCY9-RCY1)*2
    DAD    H          ;(RCY9-RCY1)*4
    PUSH   H

```

```

;
    LHLD   RCY8
    LDED   RCY2
    ORA    A
    DSBC   D
    MOV    E,L
    MOV    D,H
    DAD    H          ;*2
    DAD    D          ;*3
    POP    D
    DAD    D
    PUSH   H

```

```

;
    LHLD   RCY7
    LDED   RCY3
    ORA    A

```

FILE: CFIND.ASM

```

        DSBC      D
        DAD       H
        POP       D
        DAD       D
        PUSH      H

;
        LHLD      RCY3
        LDED      RCY4
        ORA       A
        DSBC      D
        POP       D
        DAD       D
        SHLD      RCDERV

;
;*****: DIVIDE RCDERV BY 4 TO AVOID OVERFLOW WHEN INTEGRATING IT
        LHLD      RCDERV
        SPAR      H
        RARR      L
        SPAR      H
        RARR      L
        SHLD      RCDERV

;***
;DERIV = (9-1)*4 + (8-2)*3 + (7-3)*2 + (6-4)
;
        LHLD      ABY9
        LDED      ABY1
        ORA       A
        DSBC      D
        DAD       H      ;(ABY9-ABY1)*2
        DAD       H      ;(ABY9-ABY1)*4
        PUSH      H

;
        LHLD      ABY8
        LDED      ABY2
        ORA       A
        DSBC      D
        MOV       E,L
        MOV       D,H
        DAD       H      ;+2
        DAD       D      ;+3
        POP       D
        DAD       D
        PUSH      H

;
        LHLD      ABY7
        LDED      ABY3
        ORA       A
        DSBC      D
        DAD       H
        POP       D
        DAD       D
        PUSH      H

```

-87-

FILE: CFIND.ASM

```

;
    LHLD    ABY6
    LDOD    ABY4
    ORA     A
    OSBC    0
    POP     0
    DAD     0
    SHLD    ABDERV
```

```

;*****: DIVIDE ABDERV BY 4 TO AVOID OVERFLOW WHEN INTEGRATING IT
```

```

    LHLD    ABDERV
    RRAR    H
    RARR    L
    SRAR    H
    RARR    L
    SHLD    ABDERV
```

```

;***
```

```

;RESET VALUES FOR NEXT TIME
```

```

    LHLD    RCY2
    SHLD    RCY1
    LHLD    RCY3
    SHLD    RCY2
    LHLD    RCY4
    SHLD    RCY3
    LHLD    RCY5
    SHLD    RCY4
    LHLD    RCY6
    SHLD    RCY5
    LHLD    RCY7
    SHLD    RCY6
    LHLD    RCY8
    SHLD    RCY7
    LHLD    RCY9
    SHLD    RCY8
```

```

;
    LHLD    ABY2
    SHLD    ABY1
    LHLD    ABY3
    SHLD    ABY2
    LHLD    ABY4
    SHLD    ABY3
    LHLD    ABY5
    SHLD    ABY4
    LHLD    ABY6
    SHLD    ABY5
    LHLD    ABY7
    SHLD    ABY6
    LHLD    ABY8
    SHLD    ABY7
    LHLD    ABY9
    SHLD    ABY8
```

FILE: CFIND.ASM

```
;
;
    LHLD    RCDERV
    LDED    ABDERV
    DAD     D
    SHLD    SMDERV
;
;
; COMPUTE ABSOLUTE VALUE OF RCDERV AND ABDERV, AND SMDERV
    LHLD    RCDERV
    MOV     A,H
    ANI     80H
    IF (.A,IS,NZERO)      ;THEN RCDERV < 0
        MOV     A,L
        CMA
        MOV     L,A
        MOV     A,H
        CMA
        MOV     H,A
        INX     H
        SHLD    ABSRCD
    ELSE
        SHLD    ABSRCD
    ENDIF
;
    LHLD    ABDERV
    MOV     A,H
    ANI     80H
    IF (.A,IS,NZERO)      ;THEN ABDERV < 0
        MOV     A,L
        CMA
        MOV     L,A
        MOV     A,H
        CMA
        MOV     H,A
        INX     H
        SHLD    ABSABD
    ELSE
        SHLD    ABSABD
    ENDIF
;
    LHLD    SMDERV
    MOV     A,H
    ANI     80H
    IF (.A,IS,NZERO)      ;THEN SMDERV < 0
        MOV     A,L
        CMA
        MOV     L,A
        MOV     A,H
        CMA
        MOV     H,A
        INX     H
        SHLD    ABS3MD
```

-89-

FILE: CFIND.ASM

```

        ELSE
          SHLD  ABSMMD
        ENDIF
;
;
;
        RET
;
;
;
;
;
;
GETP2:
;
;SEE IF AT LEAST 10 POINTS IN RING BUFFER
;
        LHL  RIPT
        LXI  D,20
        ORA  A
        OSBC D
        IF (PSW.IS.CARRY) ;THEN < 10 PTS
          STC
        ELSE
;
;
;SET UP INIT VALUES IN RCY1,RCY2,RCY3,RCY4
;      ABY1,ABY2,ABY3,ABY4,AND SRCOFST
;
        LXI  H,RCBUFF
        MOV  E,M
        INX  H
        MOV  D,M
        SDED RCY1
;
        INX  H
        MOV  E,M
        INX  H
        MOV  D,M
        SDED RCY2
;
        INX  H
        MOV  E,M
        INX  H
        MOV  D,M
        SDED RCY3
;
        INX  H
        MOV  E,M
        INX  H
        MOV  D,M
        SDED RCY4

```

FILE: CFIND.ASM

```
;
    INX     H
    MOV     E,M
    INX     H
    MOV     D,M
    SDED    RCY5
;
    INX     H
    MOV     E,M
    INX     H
    MOV     D,M
    SDED    RCY6
;
    INX     4
    MOV     E,M
    INX     H
    MOV     D,M
    SDED    RCY7
;
    INX     H
    MOV     E,M
    INX     H
    MOV     D,M
    SDED    RCY8
;
;
    LXI     H,ABBUFF
    MOV     E,M
    INX     H
    MOV     D,M
    SDED    ABY1
;
    INX     H
    MOV     E,M
    INX     H
    MOV     D,M
    SDED    ABY2
;
    INX     H
    MOV     E,M
    INX     H
    MOV     D,M
    SDED    ABY3
;
    INX     H
    MOV     E,M
    INX     H
    MOV     D,M
    SDED    ABY4
;
    INX     H
    MOV     E,M
    INX     H
```

FILE: CFIND.ASM

```

MOV     D,M
SDED    ABY5
;
INX     H
MOV     E,M
INX     H
MOV     D,M
SDED    ABY6
;
INX     H
MOV     E,M
INX     H
MOV     D,M
SDED    ABY7
;
INX     H
MOV     E,M
INX     H
MOV     D,M
SDED    ABY8
;
LXI     H,14      ;OFFSET FOR THE 9TH VALUE IN BUFFERS
SHLD    SRCOFST
;
LXI     H,8        ;OFFSET OF MIDDLE OF DERIV
SHLD    ROPT
;
LXI     H,0
SHLD    INTGRC
    ORA    A
ENDIF
RET
;
;
    .EXTERN CFD
    .LOC    CFD
CUROFF:    .BLKB    2
OFFTEMP:   .BLKB    2
    .RELOC
;
;
    .END
;*****:
;
;
    .IDENT    OKFIND
    .INSERT    SP80.ASM
    .INSERT    CALCOM
    .INSERT    FPMAC.SRC
    .INSERT    EFFCOM
    .INSERT    OKCOM
    .INSERT    DRVCOM
    .INSERT    SIOCOM

```

FILE: CFIND.ASM

```

        .ENTRY  OKFIND,OKGETP2
        .EXTERN TDHE
        .ENTRY  DERIV5
        .EXTERN PTRUPD
        FINIT

;
;
;ASSUME:  THAT ALL RC VALUES, AB VALUES, AND SUM VALUES
;         WILL HAVE FIRST 4 BITS = 0000H COMING FROM
;         A-D.
;
OKFIND:
;
;HL = OFFSET OF CURRENT PT ON ENTRANCE
;
; ON EXIT: IF CP FOUND, THEN CARRY SET AND
;           HL = SUM PT
;           DE = RIB CAGE PT
;           BC = ABDOMEN PT
;           TVALUE = TIME OF CP
;
        PUSH    H
        PUSH    H
        PUSH    H
        LXI     D,SPBUFF
        DAD     D
        MOV     E,M
        INX     H
        MOV     D,M
        SDED    CURSP
;
        POP     H
        LXI     D,RCBUFF
        DAD     D
        MOV     E,M
        INX     H
        MOV     D,M
        SDED    CURRC
;
        POP     H
        LXI     D,ABBUFF
        DAD     D
        MOV     E,M
        INX     H
        MOV     D,M
        SDED    CURAB
;
        POP     H
        LXI     D,TMBUFF
        DAD     D
        MOV     E,M
        INX     H
        MOV     D,M

```

FILE: CFIND.ASM

-93-

SDED CURTM

CALL GTSIGN ;CHECK SIGNS OF DERIVATIVES

```

LDA PTF
IF (.A,IS,ZERO) ;LOOK FOR MIN
    LHLD SPTMP
    LDOD CURSP
    ORA A
    DSBC D
    IF (PSW,IS,NCARRY) ;NEW TMP MIN FOUND
        LHLD CURSP
        SHLD SPTMP
        LHLD CURPC
        SHLD RCTEMP
        LHLD CURAB
        SHLD ACTEMP
        LHLD CURTM
        SHLD SMTIME

```

1. ADD CURRENT RCDERV PT TO RC COUNTS
2. ADD RCTCNT TO RC COUNTS
3. RCTCNT <= 0
4. ADD CURRENT ABDERV TO AB COUNTS
5. ADD ABTCNT TO AB COUNTS
6. ABTCNT <= 0

CALL UPDT1

```

ORA A
RET
ELSE ;POSSIBLE CP SO CHECK SLOPE BETWEEN
      ;CURRNT PT AND TEMP PT
    LHLD CURSP
    LDOD SPTMP
    ORA A
    DSBC D
    ;NOW HL HOLDS DIFFERENCE
    LDOD MINCC
    ORA A
    DSBC D
    IF (PSW,IS,NCARRY) ;THEN SMTMP IS CP
        LHLD SPTMP
        PUSH H
        LHLD RCTEMP

```

FILE: CFIND.ASM

```

        PUSH    H
        LHLD    ABTEMP
        PUSH    H
        LHLD    SMTIME
        SHLD    TVALUE
        LHLD    CURSP
        SHLD    SPTIME
        LHLD    CURRC
        SHLD    RCTEMP
        LHLD    CURAB
        SHLD    ABTEMP
        LHLD    CURTM
        SHLD    SMTIME
;
;1. SET RC COUNTS
;2. RC COUNTS <= RCTCNT
;3. RCTCNT <= 0
;4. STORE AB COUNTS
;5. AB COUNTS <= ABTCNT
;6. ABTCNT <= 0
;
        CALL    UPDT2
;
;
        POP     B        ;AB VALUE
        POP     D        ;RC VALUE
;
; SET THE APPROPRIATE BIT ON RC VALUE TO INDICATE
; A MINIMUM BRTH
        MOV     A,D
        ANI     0FH      ;FIRST 4 BITS SHOULD = 0000B
        MOV     D,A
;
        POP     H        ;SM VALUE
        STC      ;SET CARRY
        RET
    ELSE
        ;NOT A LARGE ENOUGH SLOPE
;
;1. ADD RCDERV TO RCTCNT
;2. ADD ABDERV TO ABTCNT
        CALL    UPDT3
;
        ORA     A
        RET
    ENDIF
ELSE
        ;LOOK FOR MAX
        LHLD    SPTIME
        LDOD    CURSP
        XCHG
        ORA     A
        DSBC    D

```

FILE: CFIND.ASM

```
IF (PSW,IS,NCARRY)      ;;THEN CURRENT >= SMTEMP
```

```
  LHL0    CURSP
  SHLD    SPTMP
  LHL0    CURRC
  SHLD    RCTEMP
  LHL0    CURAB
  SHLD    ABTEMP
      LHL0    CURTM
      SHLD    SMTIME
```

```
      CALL    UPDT1
```

```
      ORA A
```

```
      RET
```

```
ELSE ;POSSIBLE CP
```

```
  LHL0    SPTMP
  LDOD    CURSP
```

```
  ORA A
```

```
  OSBC    0
```

```
;NOW HL = DIFF
```

```
  LDOD    MINCC
```

```
  ORA A
```

```
  OSBC    0
```

```
IF (PSW,IS,NCARRY) ;THEN SMTEMP IS CP
```

```
  LHL0    SPTMP
  PUSH    H
  LHL0    RCTEMP
  PUSH    H
  LHL0    ABTEMP
  PUSH    H
  LHL0    SMTIME
  SHLD    TVALUE
  LHL0    CURSP
  SHLD    SPTMP
  LHL0    CURRC
  SHLD    RCTEMP
  LHL0    CURAB
  SHLD    ABTEMP
  LHL0    CURTM
  SHLD    SMTIME
```

```
      CALL    UPDT2
```

```
      POP     B      ;AB VALUE
```

```
      POP     D      ;RC VALUE
```

```
; SET THE APPROPRIATE BIT ON RC VALUE TO INDICATE
; A MAXIMUM BRTH
```

```
      MOV     A,0
```

```
      ORI     10H      ;FIRST 4 BITS SHOULD = 00018
```

```
      MOV     D,A
```

```
      POP     H      ;SM VALUE
```

-96-

FILE: CFIND.ASM

```

        STC
        RET
    ELSE        ;NOT LARGE ENOUGH SLOPE
        CALL    UPDT3
        ORA     A
        RET
    ENDIF
ENDIF
ENDIF
UPDT1:
;1. ADD CURRENT ABS(RCDERV) PT TO RC COUNTS
;2. ADD RCTCNT TO RC COUNTS
;3. RCTCNT <== 0
;4. ADD CURRENT ABS(ABDERV) TO AB COUNTS
;5. ADD ABTCNT TO AB COUNTS
;6. ABTCNT <== 0
;
        LHLD    ABSRCD
        LDA     RCINPHS
        IF (.A.IS,NZERO)        ;THEN RC INPHASE WITH SUM
            LDOD RCIN
            DAD  D
            SHLD RCIN
;
        ELSE        ;RC OUT OF PHASE WITH SUM
            LDOD RCOUT
            DAD  D
            SHLD RCOUT
;
        ENDIF
;
        LHLD    RCIN
        LDOD    RCTCNT
        DAD     D
        SHLD    RCIN
;
        LHLD    RCOUT
        LDOD    RCTOCNT
        DAD     D
        SHLD    RCOUT
;
        LXI     H,0
        SHLD    RCTCNT
        SHLD    RCTOCNT
;
;
        LHLD    ABSABD
        LDA     ABINPHS
        IF (.A.IS,NZERO)        ;THEN AB INPHASE WITH SUM

```

FILE: CFIND.ASM

-97-

```

        LDED ABIN
        DAD 0
        SHLD ABIN
;
        ELSE                                ;AB OUT OF PHASE WITH SUM
        LDED ABOUT
        DAD 0
        SHLD ABOUT
;
        ENDIF
;
        LHL0 ABIN
        LDED ABTCNT
        DAD 0
        SHLD ABIN
;
        LHL0 ABOUT
        LDED ABTCNT
        DAD 0
        SHLD ABOUT
;
        LXI H,0
        SHLD ABTCNT
        SHLD ABTCNT
;
;SUM: DO NOT SEPERATE INPHASE AND OUT OF PHASE VALUES
;      ADD THEM ALL
;
        LHL0 ABSSMD
        LDED SMIO
        DAD 0
        SHLD SMIO
;
;
        LHL0 SMIO
        LDED SMTCNT
        DAD 0
        SHLD SMTCNT
;
;
        LXI H,0
        SHLD SMTCNT
        RET
;
UPDT2:
;
;1. STORE RC COUNTS
;2. RC COUNTS <== RCTCNT
;3. RCTCNT <== 0
;4. STORE AB COUNTS
;5. AB COUNTS <== ABTCNT
;6. ABTCNT <== 0

```

FILE: CFIND.ASM

```

    LHL0 RCIN
    SHLD RCINVOL
;
    LHL0 RCOUT
    SHLD RCOUTVOL
;
    LHL0 RCTICNT
    SHLD RCIN
;
    LHL0 RCTOCNT
    SHLD RCOUT
;
    LXI H,0
    SHLD RCTICNT
    SHLD RCTOCNT
;
    LHL0 ABIN
    SHLD ABINVOL
;
    LHL0 ABOUT
    SHLD ABOUTVOL
;
    LHL0 ABTICNT
    SHLD ABIN
;
    LHL0 ABTOCNT
    SHLD ABOUT
;
    LXI H,0
    SHLD ABTICNT
    SHLD ABTOCNT
;
;SUM: DO NOT SEPERATE INPHASE AND OUT OF PHASE
    LHL0 SMIO
    SHLD SMVOL
;
;
    LHL0 SMTCNT
    SHLD SMIO
;
;
    LXI H,0
    SHLD SMTCNT
    RET
;
;
;
;
UPDT3:
;
;1. ADD ABS(RCDERV) TO RCTCNT
;2. ADD ABS(ABDERV) TO ABTCNT
;
```

```

LDA      RCINPHS
IF (A,IS,NZERO)      ;THEN INPHASE
    LHLD  ABSRCD
    LDOD  RCTICNT
    DAD   D
    SHLD  RCTICNT
ELSE
    LHLD  ABSRCD
    LDOD  RCTICNT
    DAD   D
    SHLD  RCTICNT
ENDIF

```

SUM: DO NOT SEPERATE INPHASE AND OUT OF PHASE

RET

```

LHLD    RCDERV    ;CURRENT RC DERIV VALUE
MOV     A,H
ANI     80H
STA     RCSIGN

```

```
LHLD    ABDERV    ;CURRENT AB DERIV VALUE
MOV     A,H
ANI     80H
STA     ABSIGN
```

LHLD SMDERV

```

MOV      A,H
ANI      80H
STA      SMSIGN

;
;

MOV      B,A
LDA      RCSIGN
IF (.A,EQ,.B) :THEN RC INPHASE W/ SUM
    XRA   A
    CMA
    STA   RCINPHS
ELSE
    XRA   A
    STA   RCINPHS
ENDIF

;

LDA      SMSIGN
MOV      B,A
LDA      ASSIGN
IF (.A,EQ,.B) :THEN AB INPHASE W/ SUM
    XRA   A
    CMA
    STA   ABINPHS
ELSE
    XRA   A
    STA   ABINPHS
ENDIF
RET

;
;
;
;
;
;
;
;
;
;
DERIV5:
;*****+
;DERIV USING 9 POINTS
;*****+
;
;DERIVATIVE USING 5 POINTS
;
;SRCOFST HOLDS OFFSET TO GET Y5 FROM RAW DATA BUFFERS
;
;ASSUME: RCY1,RCY2,RCY3,RCY4, ABY1,ABY2,ABY3,ABY4 ALREADY SET
;
LXI      D,RCBUFF
LHLD     SRCOFST
DAD      D
MOV      E,M
INX      H

```

FILE: CFIND.ASM

```

      MOV     D,M
      SDED    RCY9
;
      LHL D, SRCOFST
      LXI     D,ABBUFF
      DAD     D
      MOV     E,M
      INX     H
      MOV     D,M
      SDED    ABY9
;
      LXI     B,RSIZE ;UPDATE FOR NEXT TIME
      LHL D, SRCOFST
      CALL    PTRUPD
      SHLD    SRCOFST
;DERIV = (9-1)*4 + (8-2)*3 + (7-3)*2 + (6-4)
;
      LHL D, RCY9
      LD E, RCY1
      OR A
      DSBC D
      DAD H ; (RCY9-RCY1)*2
      DAD H ; (RCY9-RCY1)*4
      PUSH H
;
      LHL D, RCY8
      LD E, RCY2
      OR A
      DSBC D
      MOV E,L
      MOV D,H
      DAD H ; *2
      DAD D ; *3
      POP D
      DAD D
      PUSH H
;
      LHL D, RCY7
      LD E, RCY3
      OR A
      DSBC D
      DAD H
      POP D
      DAD D
      PUSH H
;
      LHL D, RCY6
      LD E, RCY4
      OR A
      DSBC D
      POP D
      DAD D
      SHLD RCDERV

```

FILE: CFIND.ASM

;*****: DIVIDE RCDERV BY 4 TO AVOID OVERFLOW WHEN INTEGRATING IT

```

LHLD    RCDERV
SRAR    H
RARR    L
SRAR    H
RARR    L
SHLD    RCDERV

```

;***

;DERIV = (9-1)*4 + (8-2)*3 + (7-3)*2 + (6-4)

```

LHLD    ABY9
LOED    ABY1
ORA     A
DSBC    D
DAD     H      ;(ABY9-ABY1)*2
DAD     H      ;(ABY9-ABY1)*4
PUSH    H

```

```

LHLD    ABY8
LOED    ABY2
ORA     A
DSBC    D
MOV     E,L
MOV     D,H
DAD     H      ;+2
DAD     D      ;+3
POP     D
DAD     D
PUSH    H

```

```

LHLD    ABY7
LOED    ABY3
ORA     A
DSBC    D
DAD     H
POP     D
DAD     D
PUSH    H

```

```

LHLD    ABY6
LOED    ABY4
ORA     A
DSBC    D
POP     D
DAD     D
SHLD    ABDERV

```

;*****: DIVIDE ABDERV BY 4 TO AVOID OVERFLOW WHEN INTEGRATING IT

```

LHLD    ABDERV
SRAR    H
RARR    L

```

FILE: CFIND.ASM

```

SRAR    H
RARR    L
SHLD    ABDERV

```

;***

;RESET VALUES FOR NEXT TIME

```

LHLD    RCY2
SHLD    RCY1
LHLD    RCY3
SHLD    RCY2
LHLD    RCY4
SHLD    RCY3
LHLD    RCY5
SHLD    RCY4
LHLD    RCY6
SHLD    RCY5
LHLD    RCY7
SHLD    RCY6
LHLD    RCY8
SHLD    RCY7
LHLD    RCY9
SHLD    RCY8

```

```

LHLD    ABY2
SHLD    ABY1
LHLD    ABY3
SHLD    ABY2
LHLD    ABY4
SHLD    ABY3
LHLD    ABY5
SHLD    ABY4
LHLD    ABY6
SHLD    ABY5
LHLD    ABY7
SHLD    ABY6
LHLD    ABY8
SHLD    ABY7
LHLD    ABY9
SHLD    ABY8

```

```

LHLD    RCDERV
LD      ABDERV
DAD     D
SHLD    SMDERV

```

;COMPUTE ABSOLUTE VALUE OF RCDERV AND ABDERV, AND SMDERV

```

LHLD    RCDERV
MOV     A,H
ANI     80H
IF (.A,IS,NZERO)

```

;THEN RCDERV < 0

FILE: CFIND.ASM

```
MOV    A,L
CMA
MOV    L,A
MOV    A,H
CMA
MOV    H,A
INX    H
SHLD   ABSRCD
ELSE
SHLD   ABSRCD
ENDIF

;
LHLD   ABDERV
MOV    A,H
ANI    80H
IF (.A.IS,NZERO)      ;THEN ABDERV < 0
MOV    A,L
CMA
MOV    L,A
MOV    A,H
CMA
MOV    H,A
INX    H
SHLD   ABSABD
ELSE
SHLD   ABSABD
ENDIF

;
LHLD   SMDERV
MOV    A,H
ANI    80H
IF (.A.IS,NZERO)      ;THEN SMDERV < 0
MOV    A,L
CMA
MOV    L,A
MOV    A,H
CMA
MOV    H,A
INX    H
SHLD   ABSSMD
ELSE
SHLD   ABSSMD
ENDIF
```

```
RET
```

FILE: CFIND.ASM

```
;
OKGETP2:
;
;SEE IF AT LEAST 10 POINTS IN RING BUFFER
;
    LHL    RIPT
    LXI    D,20
    GRA    A
    OSBC    D
    IF (PSW,IS,CARRY) THEN 4 10 PTS
    STC
    ELSE
;
;
;SET UP INIT VALUES IN RCY1,RCY2,RCY3,RCY4
;    ABY1,ABY2,ABY3,ABY4,AND SRCOFST
;
    LXI    H,RCBUFF
    MOV    E,M
    INX    H
    MOV    D,M
    SDED    RCY1
;
    INX    H
    MOV    E,M
    INX    H
    MOV    D,M
    SDED    RCY2
;
    INX    H
    MOV    E,M
    INX    H
    MOV    D,M
    SDED    RCY3
;
    INX    H
    MOV    E,M
    INX    H
    MOV    D,M
    SDED    RCY4
;
    INX    H
    MOV    E,M
    INX    H
    MOV    D,M
    SDED    RCY5
;
    INX    H
    MOV    E,M
    INX    H
    MOV    D,M
    SDED    RCY6
;
```

FILE: CFIND.ASM

INX H
MOV E,M
INX H
MOV D,M
SDED RCY7

INX H
MOV E,M
INX H
MOV D,M
SDED RCY3

LXI H,AB8BUFF
MOV E,M
INX H
MOV D,M
SDED ABY1

INX H
MOV E,M
INX H
MOV D,M
SDED ABY2

INX H
MOV E,M
INX H
MOV D,M
SDED ABY3

INX H
MOV E,M
INX H
MOV D,M
SDED ABY4

INX H
MOV E,M
INX H
MOV D,M
SDED ABY5

INX H
MOV E,M
INX H
MOV D,M
SDED ABY6

INX H
MOV E,M
INX H
MOV D,M

FILE: CFIND.ASM

-107-

```

;
INX      H
MOV      E,M
INX      H
MOV      D,M
SDED     ABY8

LXI      H,14      ;OFFSET FOR THE 9TH VALUE IN BUFFERS
SHLD     SRCOFST

LXI      H,8        ;OFFSET OF MIDDLE OF
SHLD     ROST

LXI      H,0
SHLD     INTOPC
ORA      A
ENDIF
RET

.END

```

108-

```

        .IDENT  PRNTVAL
        .INSERT FPMAC.SRC
        .INSERT SP90.ASM
        .INSERT CALCOM
        .INSERT LP2COM
        .INSERT SYCGM
;
        .EXTERN PRTIME
;
        .EXTERN LPON,LPOFF
        .INIT
        .EXTERN ROUND
        .ENTRY  PRNTVAL
;
; THIS ROUTINE PRINTS OUT THE DELTA VALUES:
;
;      SUM      SP      SUM/SP
;
;
; THERE ARE (2*MAXCNT) - 1 VALUES
;
;
PRNTVAL:
;
; ASSUME VALUE OF PCNT IS IN AREG ON ENTRY TO THIS ROUTINE
;
        FSTA    PCNT
; ONLY MAKE HARD COPY
GO1:
        LHLD    MAXCNT
        DCX     H
        SLAR    L
        RALR    H
        SHLD    CNT
        LIQI    4
        LXI     H,XVALUES
        SHLD    X2PTR
        LXI     H,0
        SHLD    CPPTR      ;OFFSET FOR SP,RC,AB
        XRA     A
        STA     PTF        ;THIS IS ZERO WHEN
                           ;2DD # TIME THRU LOOP
;
        XRA     A
        CMA
        STA     HRDFLG
;
        MVI     A,10
        STA     LCNT
; ONLY MAKE HARD COPY
        CALL    LPON

```

```
;OUTPUT SEVERAL LINE FEEDS
```

```
    LXI    H,CRLF$
    CALL   TXTYP
    CALL   PRTIME
    FPRN   HEAD,HOPT
    CALL   LPOFF
```

```
;
LOOP1:
```

```
;GET THE DELTA FOR SPIROMETER
```

```
    LHL    CPPTR
    LXI    D,SPMAX
    DAD    D
    MOV    E,M
    INX    H
    MOV    D,M
    SDED   SPVALUE
```

```
    LHL    CPPTR
    LDA    PTF
    IF (.A,IS,NZERO) ;THEN EVEN # THRU LOOP
        INX    H
        INX    H
```

```
    ENDIF
```

```
    LXI    D,SPMIN
    DAD    D
    MOV    E,M
    INX    H
    MOV    D,M
```

```
                                ;DE = MIN SP PT
    LHL    SPVALUE ;HL = MAX SP PT
```

```
    ORA    A
    OSBC   D
    SHLD   SPDELT ;SP DELTA
    FILA   SPDELT
    FLOT
    FLDB   CCS
    FMUL
    FSTA   SPDELT ;NOW INTERMS OF ML
```

```
;
;NOW GET SUM DELTA VALUE
```

```
    LHL    CPPTR
    LXI    D,ABMAX
    DAD    D
    MOV    E,M
    INX    H
    MOV    D,M
    SDED   ABVALUE
```

```
    LHL    CPPTR
```

FILE: PRNTVAL.ASM

-110-

```

LDA     PTF
IF (.A,IS,NZERO) ;THEN EVEN # THRU LOOP
    INX     H
    INX     H
ENDIF
LXI     D,ABMIN
DAD     D
MOV     E,M
INX     H
MOV     D,M
;DE = MIN SUM PT
LHLD    ABVALUE ;HL = MAX SUM PT
ORA     A
DSBC    D
SHLD    ABDELT ;SUM DELTA VALUE
FILA    ABDELT
FLOT
FLDB    CCS
FMUL
FSTA    ABDELT ;NOW IN ML
;
;
;NOW GET XVALUE (SUM/SP)
LDQR    X2PTR
FQWL
FSTA    XVAL
SDGR    X2PTR
;
;ONLY MAKE HARD COPY
CALL    LPON
LDA     PTF
IF (.A,IS,ZERO) ;THEN INSP
    LXI     H,ILAB
ELSE
    LXI     H,ELAB
ENDIF
CALL    TXTYP
FPRN    DELT,DELTPT
CALL    LPOFF
;
;
;
LDA     PTF
IF (.A,IS,NZERO) ;THEN EVEN # THRU LOOP
    LHLD    CPPTR
    INX     H
    INX     H
    SHLD    CPPTR
ENDIF
LDA     PTF
CMA
STA     PTF

```

DSKZ CNT,LOOP1

LP11:

CALUCULATE AND PRINT OUT THE MEAN, STD, ST ERROR, AND % ERROR

FLDB XSUM
FLDA FCNT
FDIV
FSTA MEAN

FLDB XSUM
FLDA XSUM
FMUL
FATB
FLDA FCNT
FDIV
FLDB XXSUM
FSUB
FATB
FLDA FCNT
FDIV
FSQR
FSTA STDV
FLDA FCNT
FSQR
FLDB STDV
FDIV
FSTA STERR

FLDB MEAN
FFQA 1
FSUB
FATB
FFQA 100
FMUL
FSTA PCERR
FLDA PCERR
FABS
FATB
FFQA 1
FSUB
FSTA FTEMP
FTST FTEMP,LT
IF (PSW,IS,ZERO)
FFQA 0
FSTA PCERR
ENDIF

THEN ABS(ERROR) < 1

G02:

112

```

;ONLY MAKE HARD COPY
      CALL LPON
      FPRN STATS,STPT
;OUTPUT SEVERAL LINE FEEDS
      LXI H,CRLFS
      CALL TXTYP
      CALL LPOFF

```

```

      RET

```

```

;
;
STATS: .ASCIS
Mean SUM/SP      = 303
Stdv SUM/SP      = 303
St Err SUM/SP    = 303

% ERROR          = 301 %

```

```

;
STPT: .WORD MEAN,STDV,STERR,PCERR

```

```

;
ILAB: .ASCIS '(I)
ELAB: .ASCIS '(E)

```

```

;
DELT: .ASCIS '303 303 303

```

```

;
DELTPT: .WORD ABDELT,SPDELT,XVAL

```

```

;
HOPT:

```

```

;
HEAD: .ASCIS
      Validation:
      SUM      SP      SUM/SP

```

```

;
CRLFS: .ASCIS

```

```

;
;
      .EXTERN PRT
      .LOC PRT

```

```

FTEMP: .BLKB 4

```

```

;
SPDELT: .BLKB 4

```

```

RCDELT: .BLKB 4

```

```

Y2PTR: .BLKB 2

```

```

X2PTR: .BLKB 2

```

```

ABDELT: .BLKB 4

```

113

```
LCNT: .BLKB 1
FCNT: .BLKB 4
;
MEAN: .BLKB 4
STDV: .BLKB 4
STERR: .BLKB 4
PCERR: .BLKB 4
LOOPCNT: .BLKB 1
;
.RELOC
.END
```

FILE: VSETUP.ASM

114

```

; IF VALIDATING AND IF USING SPIROBAG
; THEN SPVALUE HOLDS THE SUM MIN
      LDA      BAG
      IF (.A,IS,NZERO)      ; THEN USING SPIROBAG
      LDOD     SPVALUE      ; SUM VALUE
      JMP      L2001
      ENDIF
;
      LDOD     ABVALUE
L2001:
      MOV      M,E
      INX      H
      MOV      M,D
      ENDIF
      LDA      PTF
      CMA
      STA      PTF
      ENDIF
;
SKIP5:
;
EXIT:
;
      CALL     CSTS
      IF (PSW,IS,NZERO)
      CALL     CIMP
      IF (.A,EQ,CR) ; IF CARRIAGE RETURN
      JMP      CLKSTP
      ENDIF
      ENDIF
;
      XRA      A
      STA      INTFLG
;
;*****
      POP      PSW
      POP      H
      POP      D
      POP      B
;
      RET
;
;
;
CLKTIK:
;
      LHLD     ELPTM
      INX      H
      SHLD     ELPTM
      LDOD     SFREQ
      CALL     TDHE
      IF (PSW,IS,ZERO)      ; THEN 1 SECON

```

115

FILE: VSETUP.ASM

```

LXI    H,0
SHLD   ELPTM
LHLD   SECONDS
INX    H
SHLD   SECONDS
XRA    A
CMA
STA    SECFLG
LXI    D,60
CALL   TOHE
IF (PSW,IS,ZERO) ;THEN 1 MINUTE
    LXI H,0
    SHLD SECONDS
    LHLD ELPMIN
    INX H
    SHLD ELPMIN
ENDIF
ENDIF
RET

```

SAMPLE:

```

LDA    BAG
IF (.A,IS,ZERO) ;THEN USING SPIROMETER
    LHLD SMAD ;SUM VALUE
    MOV  A,H
    CMA
    ANI  0FH
    MOV  H,A
    MOV  A,L
    CMA
    MOV  L,A
    SHLD DATA
    LHLD RIPT
    LXI  D,ABUFF
    CALL PRNG

```

```

LHLD SPAD ;SPIROMETER VALUE
MOV  A,H
CMA
ANI  0FH
MOV  H,A
MOV  A,L
CMA
MOV  L,A
SHLD DATA
LDA  PNFLAG
IF (.A,IS,NZERO)

```

;THEN INPUTTING PNEUMOTACH-SO INTEGRATE

FILE: VSETUP.ASM

;NOTE:**** ASSUME 0 FLOW IS AT 7FFH

LHLD DATA

LXI D,7FFH

ORA A

DSBC 0

;DIVIDE BY 8 (SIGNED SHIFT RIGHT)

;CHANGE TO SHIFT ONLY 2 TIMES

;2BYTES PER INSTRUCTION

NOP

NOP

;

NOP

NOP

; SRAR H

; RARR L

SRAR H

RARR L

SRAR H

RARR L

LOED PNSUM

DAD D

SHLD PNSUM

;ADD OFFST SO ALL # > 0

LXI D,800H

DAD D

SHLD DATA

ENDIF

CALL CKFLIP

LHLD RIPT

LXI D,SPBUFF

CALL PRNG

;

ELSE ;USING FIXED VOLUME

;STORE SUM DATA IN SPBUFF SO WILL BE USED TO PICK OFF BREATHS

;

LHLD SMAD ;SUM VALUE

MOV A,H

CMA

ANI 0FH

MOV H,A

MOV A,L

CMA

MOV L,A

SHLD DATA

LHLD RIPT

LXI D,SPBUFF

CALL PRNG

;

;FILL ABBUFF WITH ZEROES

LXI H,0

SHLD DATA

LHLD RIPT

LXI D,ABBUFF

FILE: VSETUP.ASM

-117-

```

        CALL PRNG
;
;
        ENDIF
;
;FILL RCBUFF WITH ZEROES
        LXI    H,0
        SHLD   DATA
        LHLD   RIPT
        LXI    D,RCBUFF
        CALL   PRNG
;
;
;
;
;INCREMENT RIPT FOR NEXT TIME
;
        LHLD   TIME
        INX    H
        SHLD   TIME
        SHLD   DATA
        LHLD   RIPT
        LXI    D,TMBUFF
        CALL   PRNG
;
        LHLD   RIPT
        SHLD   ODRIPT
        LXI    B,RSIZE
        LHLD   RIPT
        CALL   PTRUPD
        SHLD   RIPT
;
        LDA    WRAPP
        IF (.A,IS,ZERO) ;SEE IF RIPT RESET TO 0
            LXI    D,0
            LHLD   RIPT
            CALL   TDHE
            IF (PSW,IS,ZERO)
                MVI A,OFFH
                STA WRAPP
            ENDIF
        ENDIF
        RET
;
;
;
;CKFLIP:
;
;CHECK IF NEED TO FLIP
        LDA    FLIPU
        IF (.A,IS,NZERO)

```

1

-118-

5 In practice, the Respigraph TM scaling amplifiers for the
rib cage and abdomen signals are initially set at unity gain. A
10 10 minute baseline is then taken. The program automatically
10 computes the delta values for the rib cage and abdomen signals,
discards the wild points, computes the mean values for the rib
15 cage and abdomen from the totals of the remaining delta values,
computes the standard deviations for those means, and then
computes the proportionality factor Z from Equation M. The
20 scaling amplifier for the rib cage is then adjusted to the
proportionality factor Z. The real time output from the
25 Respigraph TM representing the sum of the rib cage and abdomen
signals is then proportional to tidal volume as explained more
fully above. The program also recomputes the proportionality
30 factor Z during each subsequent 5 minute interval. If the
recomputed value differs from 1.0 by more than an acceptable
35 amount, the calibration routine can be rerun. The program also
computes the scaling factor M from Equation N if instructed to
do so upon inputting an actual tidal volume value as derived,
40 e.g., by spirometry.

While there has been shown and described herein a preferred
45 embodiment of the present invention and certain suggested
modifications thereto, it will be apparent that further changes
and modifications may be made without departing from the spirit
50 and scope of the invention. Accordingly, the foregoing
description should be construed as illustrative and not in the
55 limiting sense, the scope of the invention being defined by the
following claims.

60

1

-119-

5

Claim 1

10 1. In a method for non-invasively measuring a subject's
respiration volume of the type including providing a signal
responsive to a rib cage dimension indicative of rib cage
15 contribution to respiration volume, providing a signal
responsive to an abdominal dimension indicative of abdominal
contribution to respiration volume, multiplying at least one of
20 said rib cage and abdominal signals by predetermined weighting
factors reflecting the relative contributions of said rib cage
and abdomen to respiration volume, and summing said weighted
25 signals for providing a signal proportional to respiration
volume, the improvement comprising non-invasively determining
30 said weighting factors by

(a) totaling the delta values for said rib cage signal
35 over a baseline period of substantially steady state
breathing;

(b) totaling the delta values for said abdomen signal
40 over a baseline period of substantially steady state
breathing;

(c) dividing the average variability of the mean of
45 the total of said delta values for one of said rib cage or
abdomen

50 signals by the average variability of the mean of the total
of said delta values for the other of said rib cage or abdomen
55 signals; and

(d) and multiplying said other signal by a weighting
60 factor equaling the quotient derived from step (c).

1/3

FIG. 1

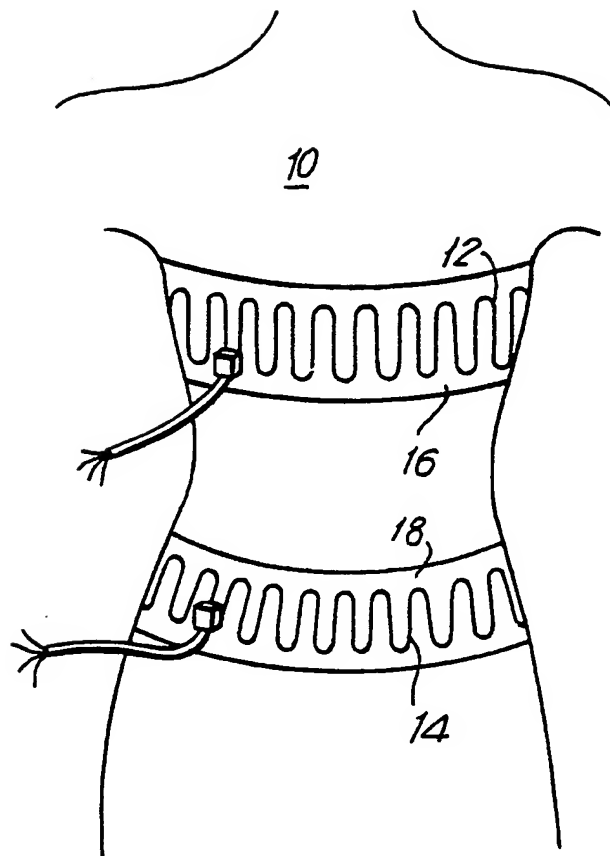


FIG. 2

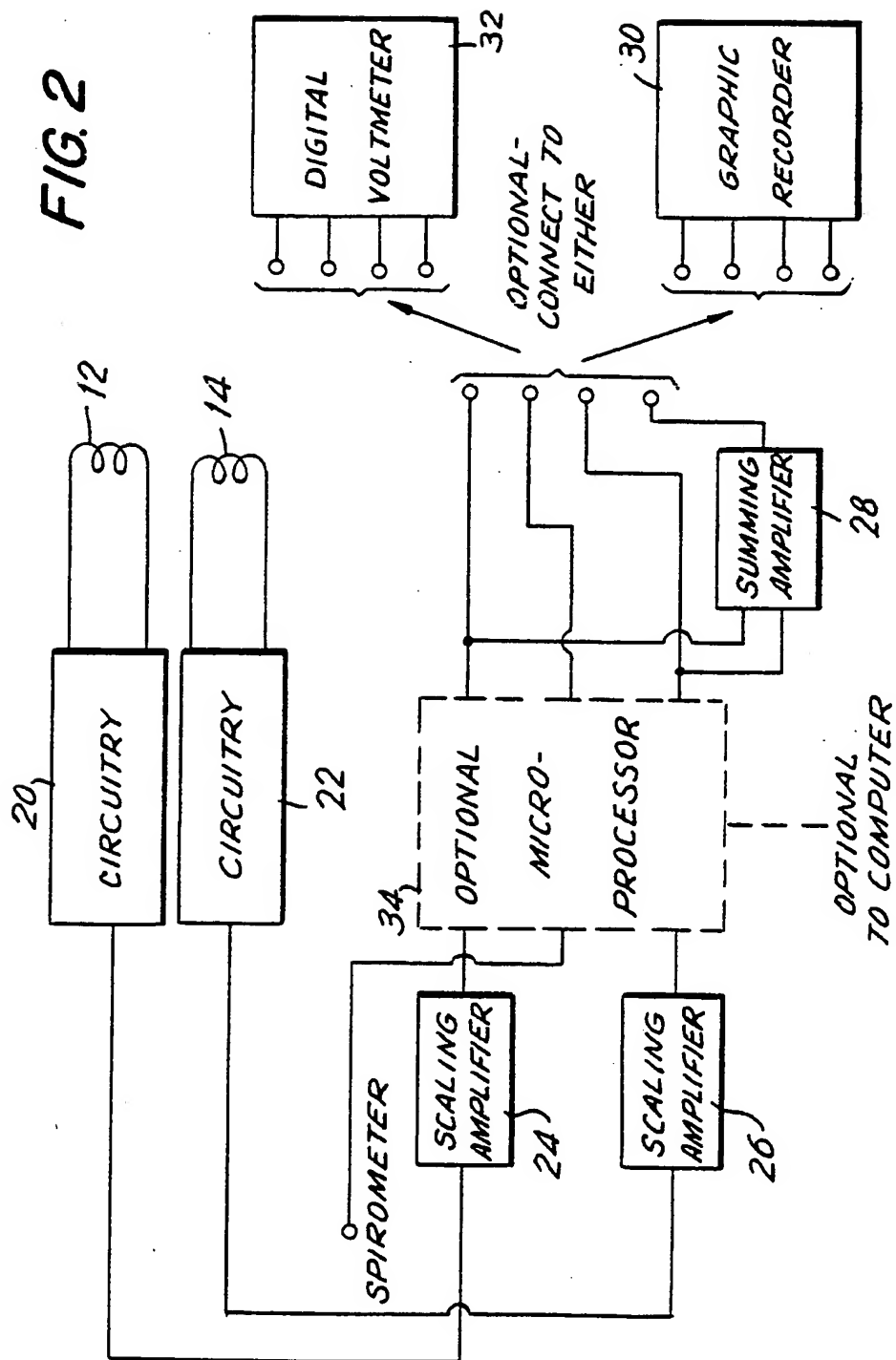
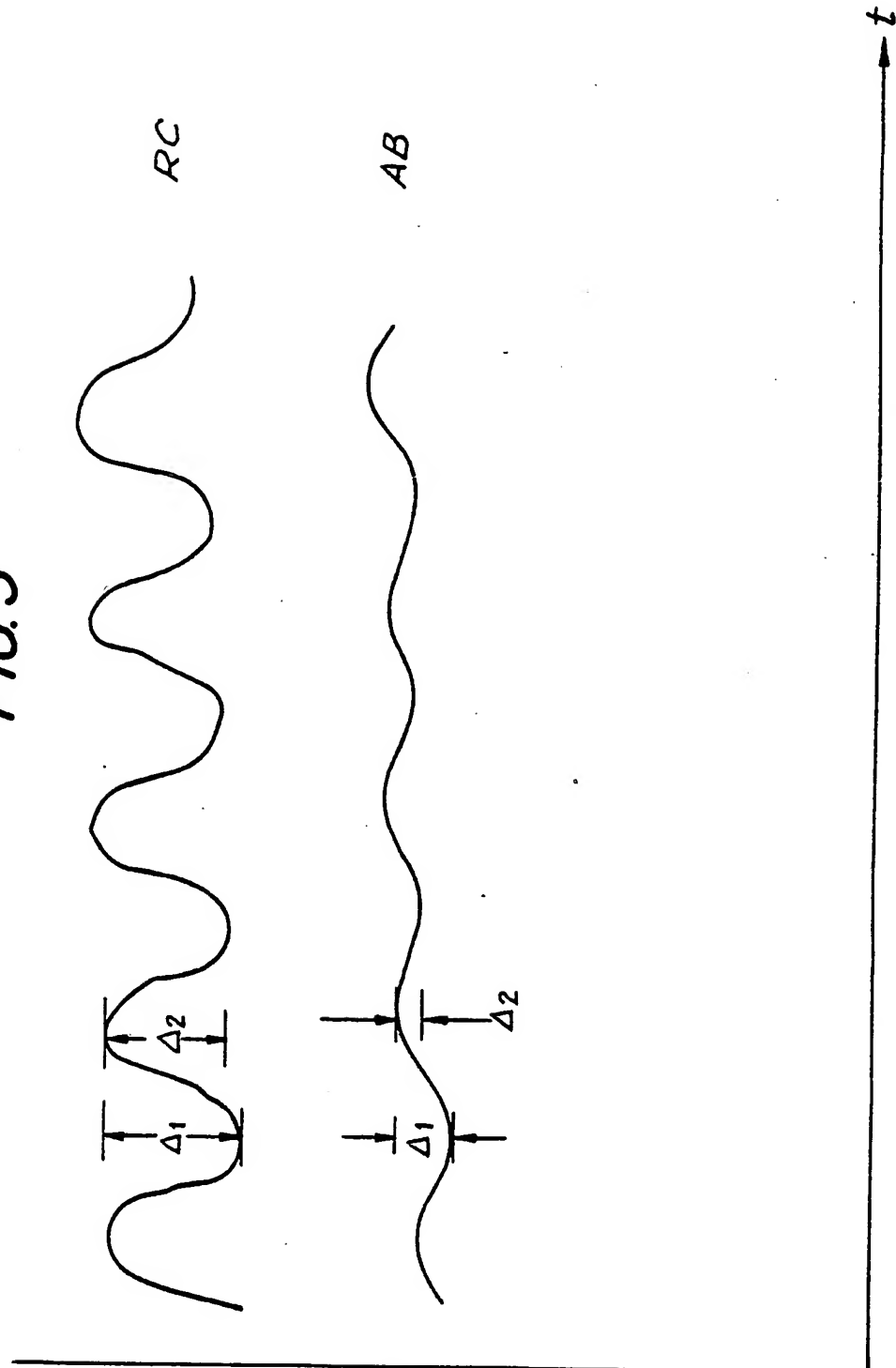


FIG. 3



INTERNATIONAL SEARCH REPORT

International Application No PCT/US87/00217

I. CLASSIFICATION OF SUBJECT MATTER (If several classification symbols apply, indicate all) ³ According to International Patent Classification (IPC) or to both National Classification and IPC IPC (4): A61B 5/08 U.S. Cl. 128/721, 725						
II. FIELDS SEARCHED <div style="text-align: center; margin-top: 10px;">Minimum Documentation Searched ⁴</div> <table style="width: 100%; border: none;"> <tr> <td style="width: 30%; border: none;">Classification System</td> <td style="border: none;">Classification Symbols</td> </tr> <tr> <td style="border: none; padding-top: 10px;">U.S.</td> <td style="border: none; padding-top: 10px;">128/716, 721, 725</td> </tr> </table> <div style="text-align: center; margin-top: 10px;">Documentation Searched other than Minimum Documentation to the Extent that such Documents are Included in the Fields Searched ⁵</div>			Classification System	Classification Symbols	U.S.	128/716, 721, 725
Classification System	Classification Symbols					
U.S.	128/716, 721, 725					
III. DOCUMENTS CONSIDERED TO BE RELEVANT ¹⁴						
Category ⁷	Citation of Document, ¹⁵ with indication, where appropriate, of the relevant passages ¹⁷	Relevant to Claim No. ¹⁸				
Y	US, A, 4,373,534 (WATSON) 15 February 1983 See the entire document.	1				
A	US, A, 4,308,872 (WATSON ET AL) 05 January 1982, see the entire document.	1				
A	US, A, 4,267,845 (ROBERTSON ET AL) 19 May 1981, see column 3, lines 21-32.	1				
<div style="display: flex; justify-content: space-between;"> <div style="width: 45%;"> <p>¹³ * Special categories of cited documents:</p> <p>"A" document defining the general state of the art which is not considered to be of particular relevance</p> <p>"E" earlier document but published on or after the international filing date</p> <p>"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)</p> <p>"O" document referring to an oral disclosure, use, exhibition or other means</p> <p>"P" document published prior to the international filing date but later than the priority date claimed</p> </div> <div style="width: 45%;"> <p>"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention</p> <p>"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step</p> <p>"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.</p> <p>"&" document member of the same patent family</p> </div> </div>						
IV. CERTIFICATION						
Date of the Actual Completion of the International Search ¹		Date of Mailing of this International Search Report ²				
25 March 1987		15 APR 1987				
International Searching Authority ¹		Signature of Authorized Officer ¹⁰				
ISA/US		John Hanley				